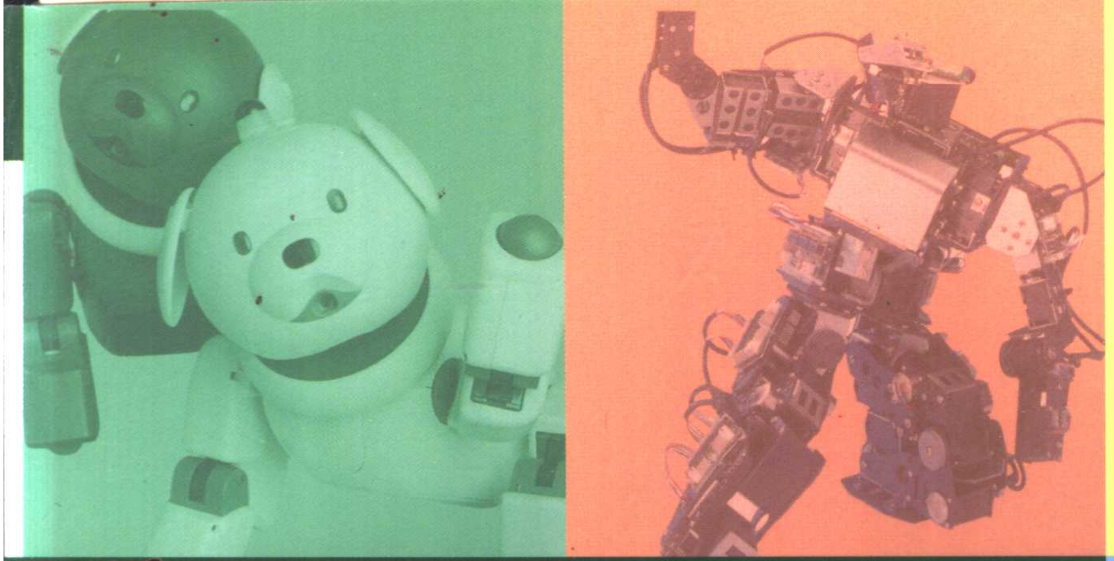
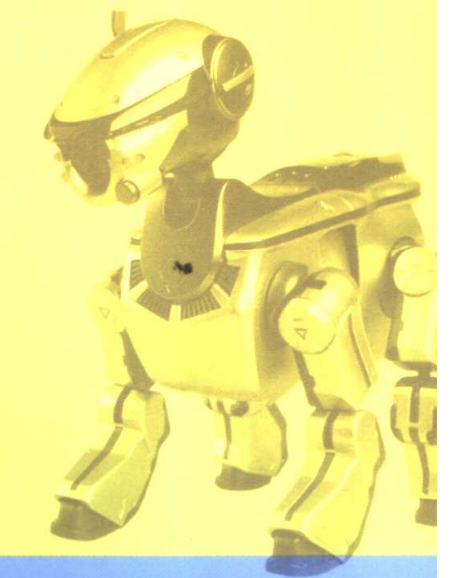
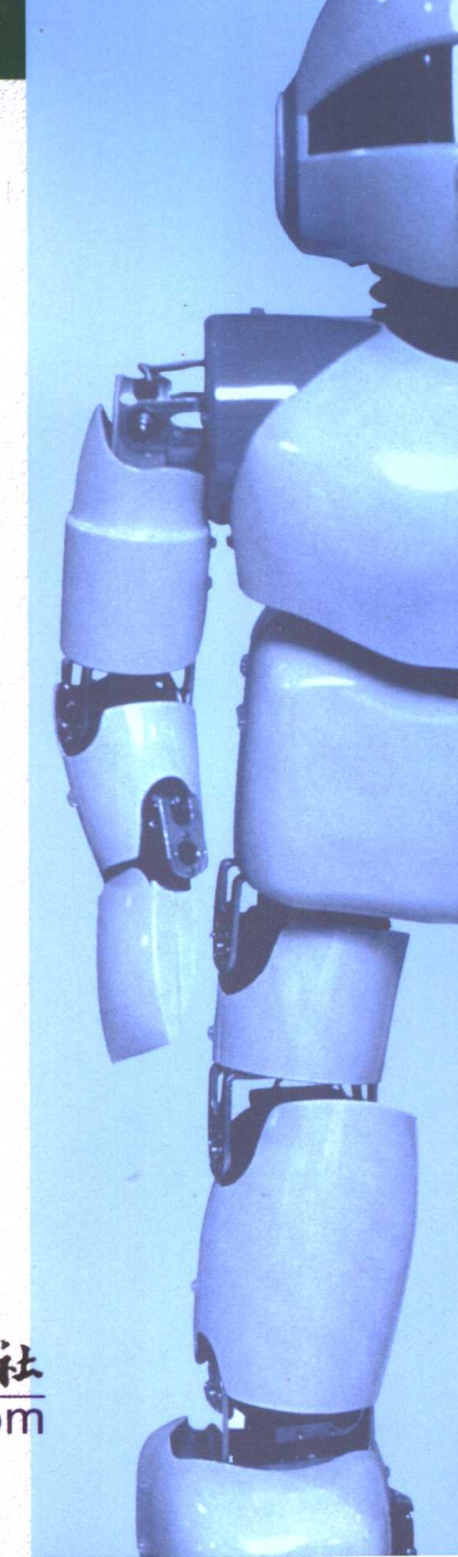
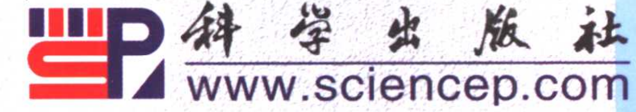




机器人创意与制作

机器人控制电子学

〔日〕船仓一郎　土屋　尧　堀桂太郎　著
宗光华　杨　洋　唐伯雁　译

机器人创意与制作

机器人控制电子学

〔日〕船仓一郎　土屋　尧　堀桂太郎　著

宗光华　杨　洋　唐伯雁　译

科学出版社

北京

图字：01-2004-2581 号

内 容 简 介

本书是机器人创意与制作系列之一。本书从制作机器人所需的电子元器件及相关的基础知识入手，主要介绍微控制器的基础知识，直流电机和步进电机的结构、基础知识、控制方法，传感器的基础知识、常见传感器的工作原理和使用方法，巡线小车和遥控坦克的制作方法，几种机器人竞赛的集锦等。本书简明易懂，重点突出，引用大量实例，可以开阔读者的眼界。

本书可作为大专院校理工科学生补充机械电子学、机器人工程、人工智能、计算机控制等领域知识的参考书，还可作为高中学生的课外科技活动辅导教材。

图书在版编目(CIP)数据

机器人控制电子学/(日)船仓一郎等著；宗光华等译. —北京：科学出版社，2004
(机器人创意与制作)
ISBN 7-03-013168-1

Ⅰ.机… Ⅱ.①船…②宗… Ⅲ.机器人控制-电子学 Ⅳ.TP24

中国版本图书馆 CIP 数据核字(2004)第 033032 号

责任编辑：杨 凯 崔炳哲 / 责任制作：魏 谨
责任印制：刘士平 / 封面设计：李 祥

科学出版社 出版
北京东黄城根北街16号
邮政编码：100717
http://www.sciencep.com
源海印刷有限责任公司印刷
北京东方科龙图文有限公司 制作
http://www.okbook.com.cn
科学出版社发行 各地新华书店经销

*

2004 年 5 月第 一 版 开本：B5(720×1000)
2004 年 5 月第一次印刷 印张：14 1/2
印数：1—5 000 字数：245 000

定 价：28.00 元

(如有印装质量问题，我社负责调换〈新欣〉)

▲▲▲▲▲▲▲▲▲▲▲▲▲ 中译本序

前不久，美国火星探测器“勇气”号和“机遇”号这对孪生兄弟在先后经历了半年多、数亿公里的星际旅行后终于成功地着陆在目的地。众所周知，真正踏上火星表面，作为首位地球使者漫步火星的巡游车，实际上是两个机器人。不过它们两个非同一般。一则，它们的长相简直“帅呆了”：6个独立悬挂的车轮，能在崎岖的火星表面坚实地前进；太阳能电池翻板的双翼和竖立的360°全景摄像机了望台，犹如一只展翅欲飞的鹤；而与人体上肢结构类似的机械臂，一旦到达目标附近就轻轻地舒展开去采集火星表层标本。另一则，它们堪称“目前世界上最聪明的机器人”和一座功能完美的移动实验室，总共携带了7类科学探测仪器，把星际探测器的自动化程度提高到前所未有的水平。

火星巡游机器人的闪亮登场掀起机器人爱好者的又一波热潮，特别是激发了一批青少年创意和制作机器人的兴趣与灵感。因此在这里，向读者推荐一套有关机器人创意与制作的参考书——“机器人创意与制作”系列。该系列共有4册，分别是：

(1)《机器人控制电子学》

撰写本书的三位作者长期从事日本工业专科学校机械工程、电子技术和电气信息工程等课程的教学。此书是本套丛书中较为通俗的入门篇，以高中学生和低年级大学生为对象是本书的特色。本书以几种较简单的机器人为主线，重点介绍所涉及到的机器人控制电子学知识(如控制器、传感器、驱动器、电机控制等)，内容相当简单扼要，而且注重基础和制作的实践。

(2)《机器人设计与控制》

《机器人设计与控制》和《机器人控制器与程序设计》均原属于McGraw-Hill公司出版的《TAB Electronics Robot DNA Series》系列中的分册。撰写本书的两位作者丹尼斯·克拉克和迈克尔·欧文斯既是相关方面的行家里手，也是身处第一线的实践者。

本书重点讲解如何搭建一个简易的机器人系统的物理实体，即机构、传动、驱动与控制。

(3)《机器人控制器与程序设计》

撰写本书的作者迈克·普瑞德科是机器人控制和程序设计方面的专家。在该书中作者以微芯公司的 PICmicro 系列微控制器为核心，相当详细和深入浅出地介绍了设计机器人控制器软件开发工具，软件编程和调试的基本过程，提供了有实用价值的接口程序和控制应用代码。

《机器人设计与控制》和《机器人控制器与程序设计》不涉及高深难懂的理论，也没有复杂的数学公式，但是基础门槛仍然稍高，需要具备一定的物理、计算机、电子学方面的基础知识，适合初步掌握了 C 语言编程技巧的读者阅读和实践，一般作为大学高年级，或研究生低年级学生的参考书。

(4)《小型机器人的基础技术与制作》

本书作者高桥友一、秋田纯一、渡边正人从事电子与信息方面的教学工作，长期参与 RoboCup 小型机器人组的活动。相对来说《小型机器人的基础技术》与制作是本系列难度较高的一册，因为 RoboCup 小型机器人组比赛本身对抗性强，对性能指标的要求就较高，又有多机器人协调、全局视觉和无线电收发等技术关键。此外投资硬件平台的经费高。

目前机器人教科书都不大适合初学者。本系列基本上属于普及层次的读物，各分册程度略有差别，内容互为补充。它们所针对的是一批高中和大学的机器人爱好者。初学者学习机器人，最好的入门途径是从自己创意和制作简易机器人起步，有了实践的感受和体验，对日后学习理论的重要性会有更深刻的认识和更强的目的性。如果读者愿意自己尝试一下，那么不妨来根据自己的具体条件选择其中合适的一二册来读一读。

由于译者的水平所限，书中难免存在错误和缺点，恳请批评指正。

让思维沸腾起来，让智慧行动起来！

开拓创造力，激发想像力，锻炼实际动手能力！

宗光华

北京航空航天大学机器人研究所

译者的话

机器人是当代高新技术综合的产物，涉及机械、电子、传感器、自动化、计算机、信息处理等多门学科。多年来，不仅机器人学始终处于世界学术阵地的前沿，而且机器人作为一种学习载体逐渐被人们所广为利用。最近，在我国中学，模型机器人开始成为素质教育、技能实践的选题之一，各种机器人赛事正方兴未艾。

正如大脑是人类的灵魂和指挥中心，把控制系统称为机器人的大脑也毫不为过。机器人的感知、判断、推理都是通过控制系统的输入、运算、输出来完成的，所有行为和动作都必须通过控制系统发出相应的指令实现。可见控制系统就是机器人智慧的源泉。

本书围绕机器人控制系统这个核心，介绍了搭建一个机器人控制系统所必需的电子学基础知识和相关的电子元器件。本文始终围绕一台简单的机器人——巡线小车的控制系统的建立展开，使读者能够做到学以致用，从而培养起信心和兴趣。本书的三位作者船仓一郎、土屋尧、堀桂太郎是长期工作在第一线的学者，有丰富的教学经验。

全书的内容分为八章。

第1章简要介绍制作机器人所需的电子元器件及相关的基础知识。第2章详细说明机器人制作的核心控制元件——微控制器的基础知识，并较多地涉及到PIC和H8两种常用的微控制器的相关知识。这一章是本书的难点和重点，也是进一步学习后续各章的基础。第3章和第4章分别介绍两种机器人的典型驱动元件——直流电机和步进电机的结构、基础知识，以及控制方法。第5章概要地讲述传感器的基础知识、几种常用传感器的工作原理和使用方法。第6章讲解红外线遥控装置的工作原理和编程方法，以及提高抗干扰能力的措施。第7章介绍巡线小车和遥控坦克的制作方法。第8章是现时在日本颇为流行的几种机器人赛事的集锦，使读者的眼界更加开阔。

本书的一个特点是浅显易懂，没有深奥的理论推导和数学公式，内容非常实用。作者在介绍微控制器这个难点上处理得非常得体，能抓住最核心的内容，而将面向高级用户的功能予以省略（当然有兴趣的读者可参考相关书籍）。书中还引用大量的实例，比较适合我国高中学生的知识结

构和层次，即使是初学者通过学习也能领会原理并学会应用。

本书的第二个特点是既有针对性又有普遍性。全书的内容将巡线小车的控制系统制作贯穿始终，所介绍的相关基础知识及基本元器件又同样适用于其他模型机器人的制作。运用这些知识，读者能举一反三地进行其他机器人的创意设计与制作活动，充分体现了作者在内容安排上的良苦用心。

本书的最后一章为读者提供了一个了解当今世界上各种机器人竞赛赛事的窗口，可以开阔读者的眼界。在可能的情况下，有兴趣的读者不妨自己动手亲身体验一下。

有些机器人爱好者在制作之前可能尚对机器人知之甚少，因此本书可作为大专院校理工科学生补充机械电子学、机器人工程、计算机控制等领域知识的参考书，还可作为高中学生的课外科技活动的辅导教材，也可作为辅导教师的参考书。通过自学本书，大学低年级同学可以获得一些机器人控制技术方面的知识。

由于译者水平有限，书中难免会有缺点和错误，恳请读者批评指正。

▲▲▲▲▲▲▲▲▲▲▲▲ 前　　言

大多数人从少男少女时代就憧憬着是否有机器人存在，同样，当笔者们一谈到机器人这样的话题，也不由得心情激动。事实上，目前机器人的存在不仅已经成为现实，而且广泛地进入到工厂生产线、保安、康复及家务等许多领域。近年来，宠物机器人更是受到人们空前的喜爱。

于是，很自然地，许多人对自己制作机器人都跃跃欲试。幸运的是，制作机器人不可缺少的各种电子元器件的功能已经足够强大、价格也相当低廉，从而使爱好者们能够越来越方便、简单地实现自己设计和制作机器人的梦想。

尽管如此，对于动手实际制作机器人的初学者来讲，这一过程所涉及到的传感器、微控制器、电机控制等与电子学有关的知识还是令人大伤脑筋的。因此，本书的目的在于，使初学者在一边对相关电子学基础知识学习理解的同时，一边能够实际动手制作机器人。当然，本书中所介绍的有关电子学的知识，还可用于各种不同种类的机器人的制作或参考。

作者建议初学者们在本书的引导下能独立地面对制作机器人的挑战，如果读者在学习本书心得的基础上哪怕是制作出一台能动作的机器人，作者都感到非常荣幸。

本书中所采用的各种元器件、照片等，得到了相关公司、销售商、委员会及各种团体的鼎力支持和帮助，借这个机会表示衷心的感谢。

作　者

本书中所涉及的程序基于免费获取的 C 编译器(PICC Lite)。同时，附录中收录了相应的汇编语言编写的程序。而与红外线遥控相关的程序，由于需要精确地计算处理时间，所以，本书中仅给出了它们的汇编语言程序。

本书所涉及到的各种源程序，可以从下面的主页下载：

① OHM 社的主页—— http://www.ohmsha.co.jp

② 作者的主页——http://www.akashi.ac.jp/cnt-b_gakka.html

→电气情报工学科→教师及其授课科目绍介→堀桂太郎→下载

目　　录

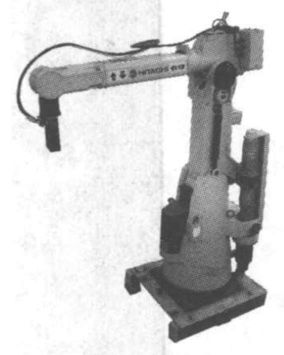
第1章

机器人控制基础

在电视的广告中人们经常可以看到双足步行机器人像人一样灵巧地行走，那么，它的动作是怎么实现的呢？实际上，它是把各种各样的元器件通过巧妙的组合来实现各种动作的。一般情况下，要控制机器人必须有各种各样的电路，例如驱动电机等动力源动作的电路，检测外界变化的传感器电路，以及针对上述电路进行信息处理、控制的 CPU(相当于人类的大脑)等。此外，还必须掌握机器人整体平衡的控制程序方面的知识。这一章里将讲解对后续各章节内容的理解所必需的机器人各部分的组成、微控制器的基础以及有关电子元器件的基础知识。

1.1 各种各样的机器人

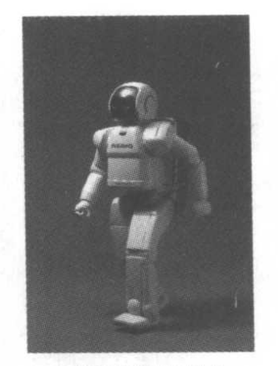
一提到机器人，留给人们印象最深的是铁臂阿童木或广告中出现的双足步行机器人。实际上，在微型计算机发达的当今，人类社会的现实中活跃着各种各样的机器人，如工业生产中使用的工业机器人及水下、洁净室、高空等人类不便到达的危险场所作业的机器人，此外，还有娱乐机器人、相扑比赛机器人等(参见图 1.1)。本书重点介绍有关控制这些机器人的电子学知识。



工业机器人
(兵库县兵库职业高中实习室日立制作所制造)



娱乐机器人



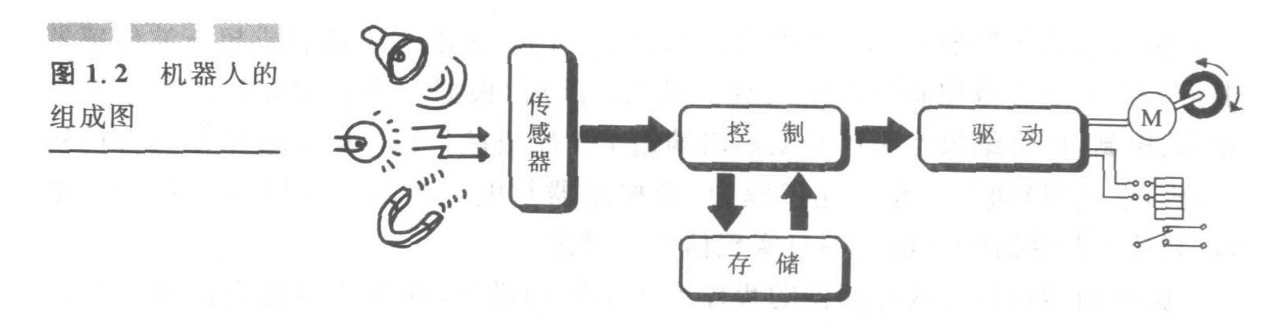
双足步行机器人
(本田技研工业株式会社)

图 1.1　各种各样的机器人

1.2 各部分组成

机器人一般由图 1.2 所示的几个部分组成，包括感知光、声音、温度、电磁等环境状况的**传感器**部分；存储信息并进行计算、判断，决策下一步动作的控制部分；以及将结果输出到电机或**继电器**，再执行手、足或车轮的实际动作的驱动部分。

图 1.2　机器人的组成图

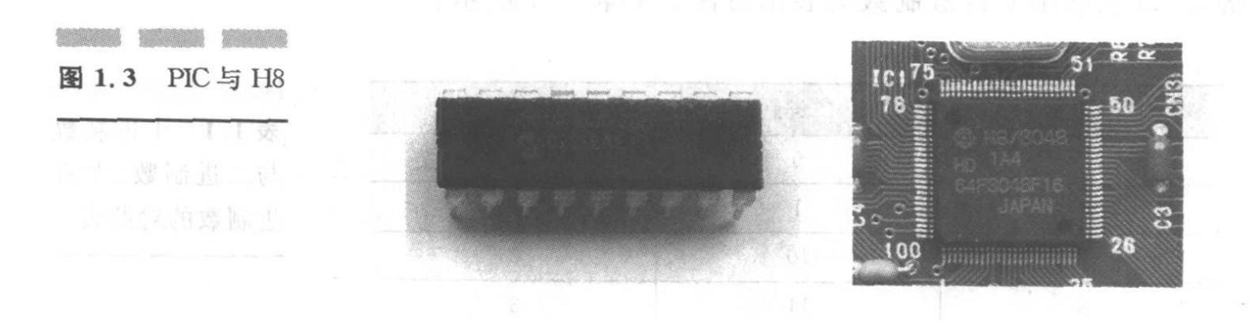


1.3　微控制器基础

机器人控制一般采用高性能的计算机，而小型机器人的控制则经常使用集计算机的基本功能于一体的IC封装芯片即**LSI(大规模集成电路)**，这种**IC芯片**称为**微型计算机**，也常常简称为**微控制器**，或称为**单片机**。

目前有许多种类的微控制器，本书主要以图 1.3 所示的**微芯(Microchip technology)公司**生产的 PIC 与 **Renasas technology 公司**的 H8 为例进行说明(详见第 2 章)，它们经常被用于控制中。

图 1.3　PIC 与 H8



微控制器的组成与一般的计算机相同，其动作也由**程序**来完成。图 1.4 所示的微处理器有**输入**、**输出**、**存储**、**运算**及控制功能，输入、输出仅由端口出入。在存储器以外还可以追加外部存储器。图 1.4 的虚线所包围的部分相当于一个微控制器。

图 1.4　微控制器的功能

运　算
控　制
输　入
存　储
输　出
外部存储

微控制器在机器人中完成的主要功能是:首先,把由光传感器或超声波传感器取得的信号通过微控制器的输入端口读入。然后,根据存储器所存储的程序进行运算、**控制**,再将结果作为信号从输出端输出。输出信号通过电子电路使执行机构(电机,继电器)动作。在上述过程中,微控制器与电气电路之间的桥梁被称为**接口**,其任务是通过输入输出端口实现信号的进出。

微控制器根据写入存储器的程序产生不同的动作,而程序则是根据微控制器内部的"0"和"1"所组合成的**二进制数**进行操作的。在电路中,二进制数"1"表示高电压状态,"0"表示低电压。例如,在图1.5中,开关ON表示"1",OFF表示"0"。

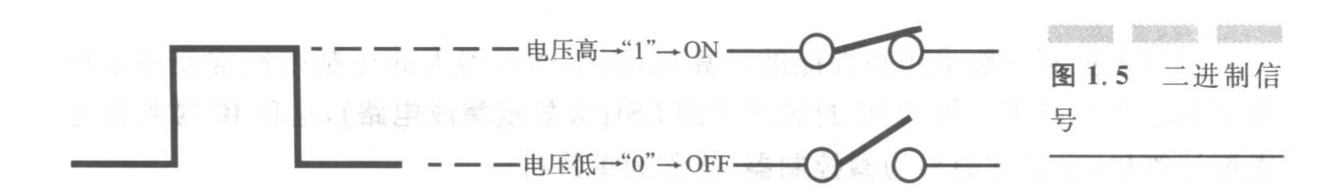


图1.5 二进制信号

换句话说,可将开关的状态变化表示为"0"和"1"两种方式,计算机通过许多种开关的组合来表示程序。可是,二进制数比通常使用的**十进制数**位数多,不易理解,所以也常用**十六进制数**的表示方法。如表1.1所示。

表1.1 十进制数与二进制数、十六进制数的对照表

十进制	二进制	十六进制
0	0	0
1	1	1
2	10	2
3	11	3
4	100	4
5	101	5
6	110	6
7	111	7
8	1000	8
9	1001	9
10	1010	A(a)
11	1011	B(b)
12	1100	C(c)
13	1101	D(d)
14	1110	E(e)
15	1111	F(f)

为了便于理解微控制器的动作原理，下面考虑图 1.6 所示的**巡线小车**。小车的主要组成部分是两个安装在小车前部的检测直线的传感器和安装在后轮的两个驱动电机。

图 1.6 巡线小车

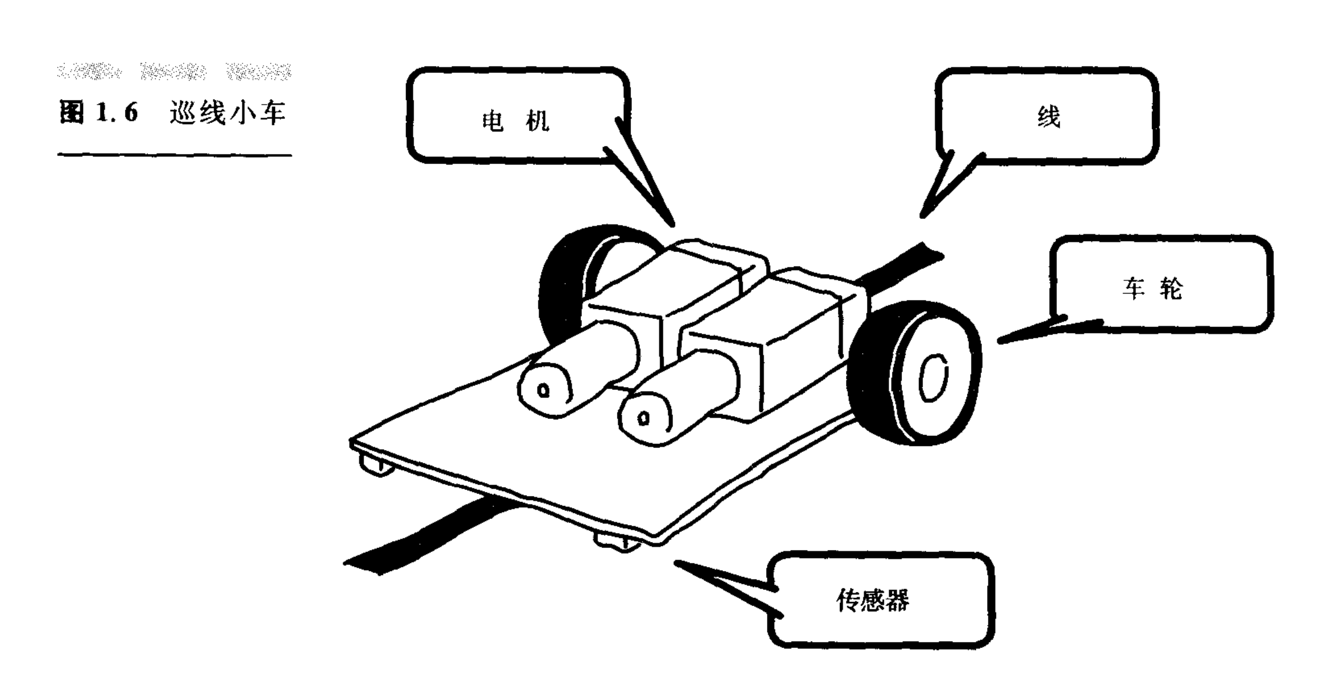
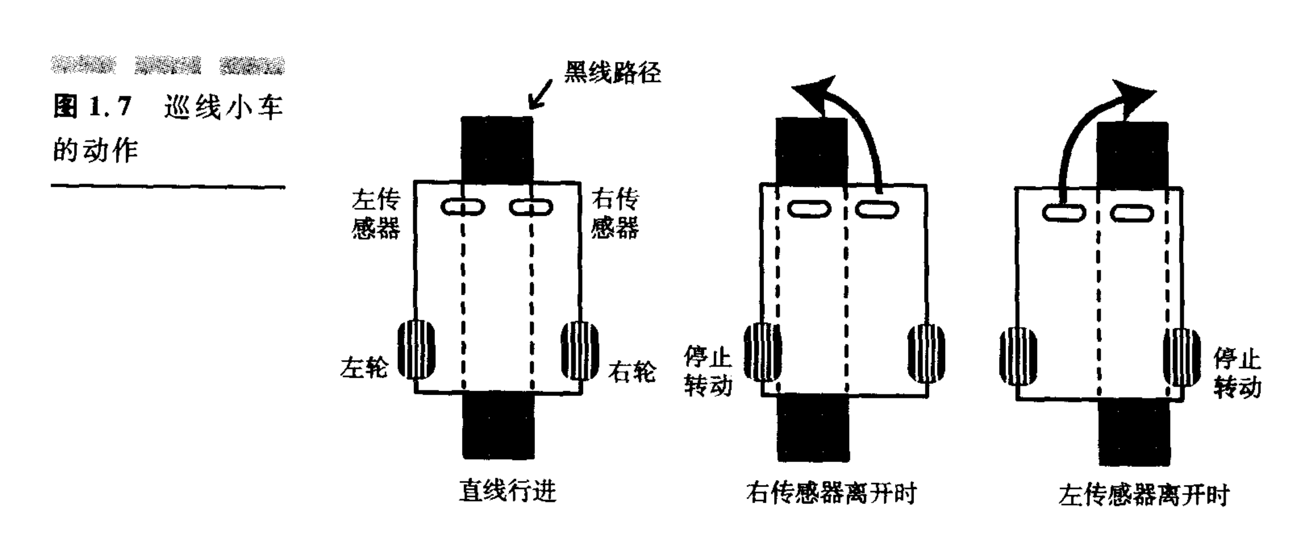


图 1.7 表示的是，若两个传感器都在直线的上方，则可以得到由传感器获取(有直线 on(1))的信号。根据这个信号，驱动后轮的两个**电机**(输出为 1)，小车继续前行。

图 1.7 巡线小车的动作



若某一侧传感器检测不到路线，则该侧对应传感器的输出信号为“没有直线 OFF(0)”。这时，该传感器相对一侧电机的转动应被停止(输出 0)，以便对方向进行校正。要实现这个动作，就需要编写相关的程序。

1.4 电子元器件的基础知识

用于机器人控制的电子元器件主要有以下几种：电阻、电容器、线圈、晶体管、IC 芯片、传感器及微控制器等。

下面就逐个地介绍它们的动作原理及使用方法。关于微控制器的知识将安排在第 2 章做详细的解说。

1.4.1 电 阻

顾名思义，电子电路中电阻具有阻止电流通过的性质，所以电阻的主要功用就是阻止电流或电压。电阻值的单位为 Ω(欧)。电阻的种类很多，如炭膜电阻、固体电阻、绕线电阻、集成电阻等等，另外，还有可改变阻值的**可变电阻器**，它常常作为**分压器**在电路中起电压分压的作用。图 1.8 是电阻与它的表示符号。

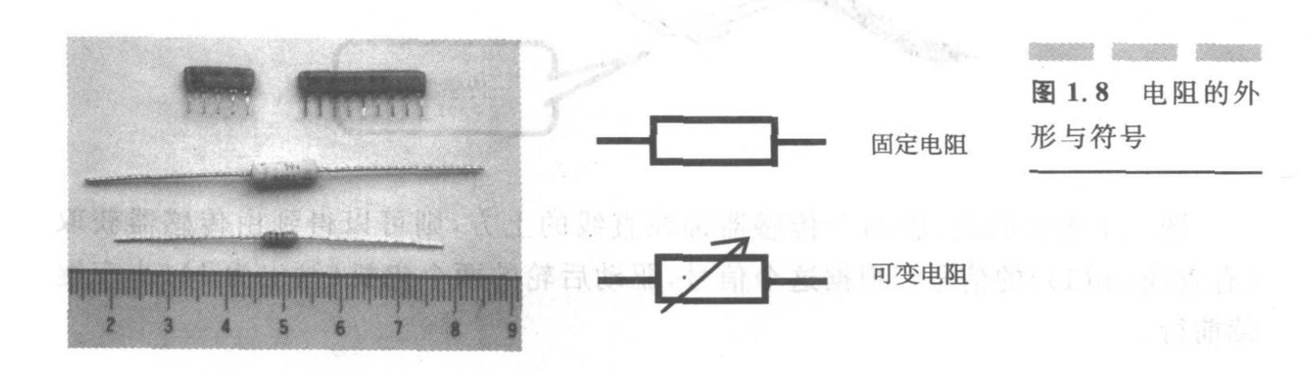


图 1.8 电阻的外形与符号

人们熟知的欧姆定律指出，给电阻两端施加电压，电路上就会有与其电压大小成正比的电流流过，如图 1.9 所示。

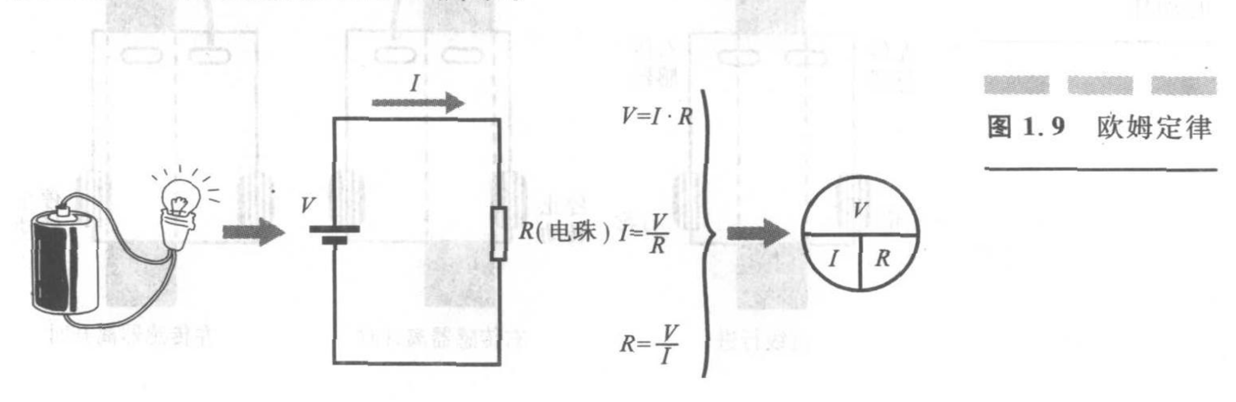
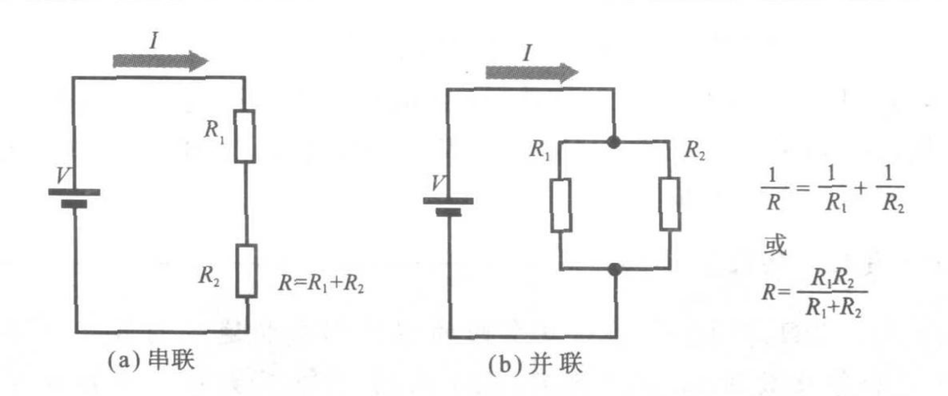


图 1.9 欧姆定律

把电阻串联接入电路或并联接入电路时的合成电阻 R 如图 1.10 所示。

图 1.10 电阻的串联与并联

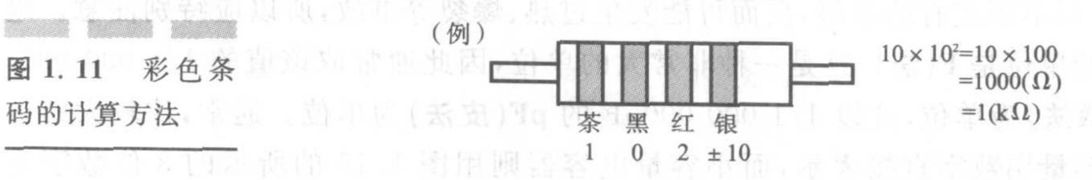


电阻的大小的表示方法是阻值的较大的电阻直接将阻值印在电阻表面，而阻值较小的电阻由于书写困难，一般用表 1.2 所示的**彩色条码**来表示。

表 1.2 电阻阻值的彩色条码（E24 系列）

颜 色	第 1 颜色带	第 2 颜色带	第 3 颜色带	第 4 颜色带
	第 1 数字	第 2 数字	幂 次	误 差
黑	0	0	10 的 0 次幂	—
茶	1	1	10 的 1 次幂	±1%
红	2	2	10 的 2 次幂	±2%
橙	3	3	10 的 3 次幂	—
黄	4	4	10 的 4 次幂	—
绿	5	5	10 的 5 次幂	±0.5%
蓝	6	6	10 的 6 次幂	±0.25%
紫	7	7	10 的 7 次幂	±0.1%
灰	8	8	10 的 8 次幂	—
白	9	9	10 的 9 次幂	—
金黄	—	—	10 的－1 次幂	±5%
银白	—	—	10 的－2 次幂	±10%
无色	—	—	—	±20%

图 1.11 彩色条码的计算方法



这个例子中，电阻值为1kΩ时，其误差为±10%(900～1100Ω)。在选择电阻时，不仅要注意电阻值的大小，而且也要注意流过电流的发热效果($P=I^2R$(W))。炭膜电阻多为1/2W、1/4W、1/8W的小型电阻，如果电路中有大电流通过，必须使用瓦数较大的电阻。

专栏：电阻的系列

120Ω、330Ω、5.1kΩ等电阻的阻值都是非连续的数字。实际上阻值基本上符合等比数列，如E12系列、E24系列、E96系列等。根据这些数字的组合能很容易地知道它们的阻值。

表1.3 电阻阻值的系列

E12系列	10	12	15	18	22	27	33	39	47	56	68	82
E24系列	10	11	12	13	15	16	18	20	22	24	27	30
	33	36	39	43	47	51	56	62	68	75	82	91

1.4.2 电容器

电容器这种电子元件具有阻止直流而让交流通过的性质。此外，对应于直流又具有储存电荷的功能。从图1.12的原理上看出，电容器由两块金属板之间加入**绝缘介质**而组成，**介质**研究的进展，使电容器的体积越来越小，容量越来越大。

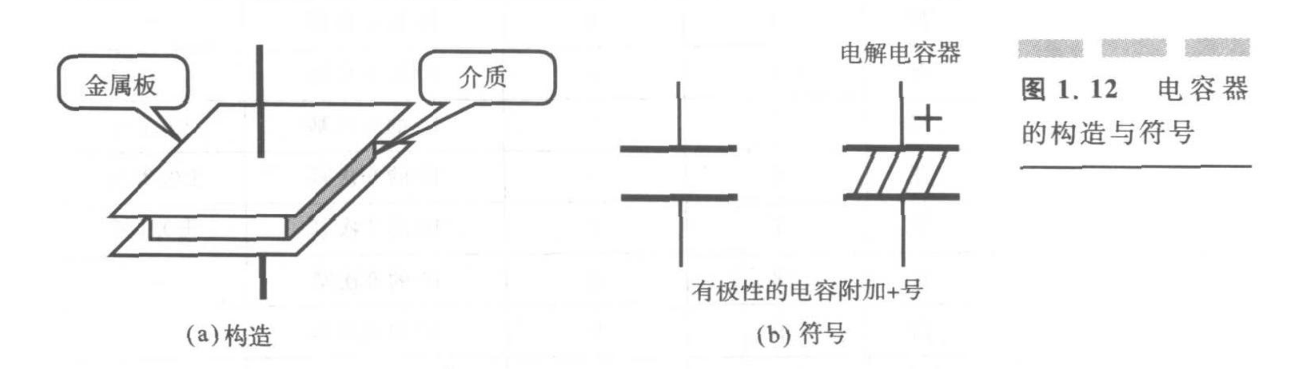


图1.12 电容器的构造与符号

作为电子元器件使用的电容器，根据其材料不同分为陶瓷电容器、云母电容器、电解电容器、钽电容器、树脂电容器等，此外也有能改变容量的**可变电容器**。图1.13中给出了几种电容器的例子。图1.14的电解电容器有极性，若反接在电路中，不仅得不到应有的容量，反而可能发生过热、爆裂等事故，所以应特别注意。电容容量的单位是**F(法)**，这是一种非常大的单位，因此通常取该值的1/1 000 000，即**μF(微法)**为单位，或以1/1 000 000μF的**pF(皮法)**为单位。通常，比较大的电容器的容量用数字直接表示，而小容量电容器则用图1.15的所示的3位数字表示。

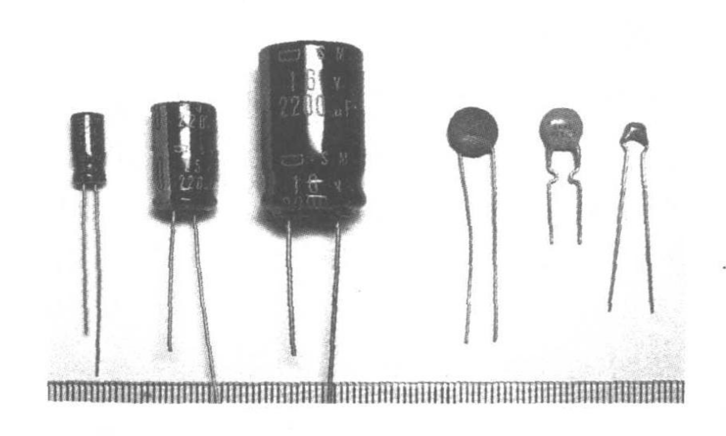


图 1.13　几种电容器的例子

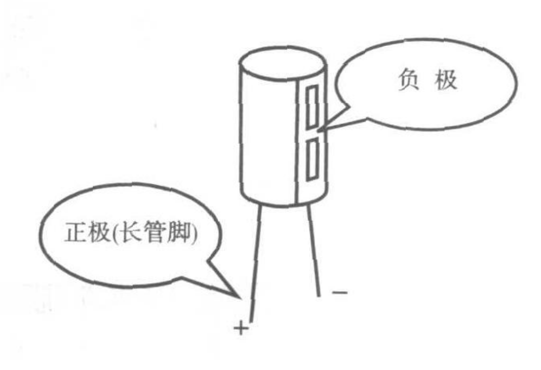
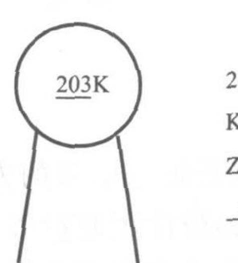


图 1.14　电解电容器的极性



203=20 × 10^3(pF)=0.02(μF)

K: 容许误差 ± 10%

Z: 容许误差+80%~-20%

_: 表示额定电压50V

图 1.15　小容量电容器的读法

电容的串联与并联的总容量计算如图 1.16 所示，很显然，计算公式与电阻相反。

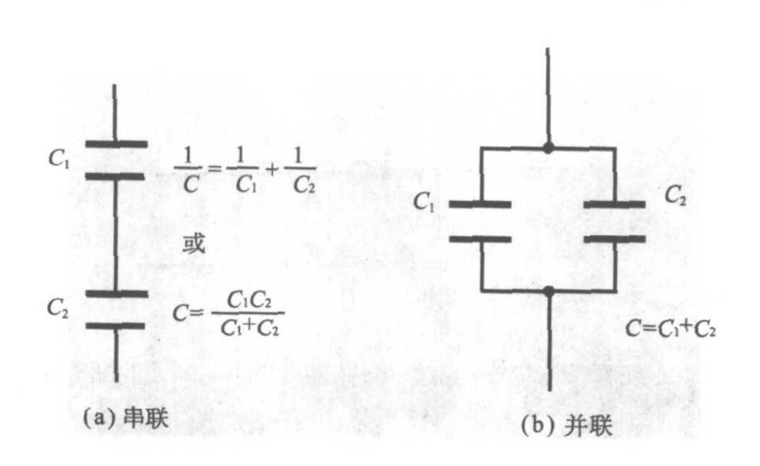


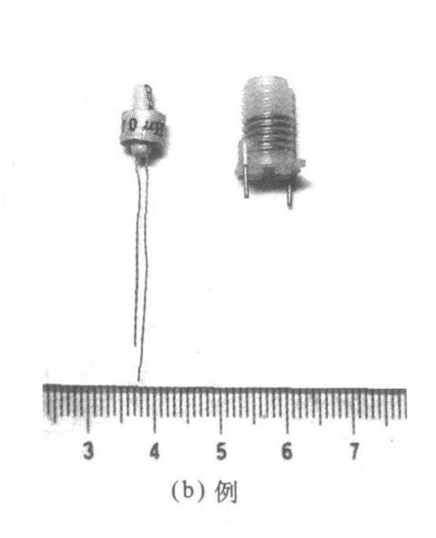
图 1.16　电容的串联与并联

1.4.3 线　圈

线圈的性质是通过直流容易，而阻止交流流动。它可以储存电磁能量，也可用作开关稳压器。虽然机器人中的线圈不起这些作用，但是在电机及继电器中经常使用线圈，若不知道线圈的上述性质，设计的电路就可能无法正常工作，导致元器件的故障等等。线圈的符号表示如图 1.17 所示。



(a) 符号



(b) 例

图 1.17　线　圈

1.4.4 二极管

二极管是一种半导体，可根据所施加电压的方向，产生通电或不通电的效果，因此它可以作为整流器件。它的整流作用通过 **N 型半导体**(半导体中的电子携带电流流动)和 **P 型半导体**(电子迁移所剩空穴携带电流流动)的结合而实现，也称为 **PN 结**。根据半导体的用途和材料，二极管可分为整流二极管、开关二极管、稳压二极管、发光二极管等。图 1.18 给出了二极管的例子。图 1.19 给出了二极管的表示符号及构造。

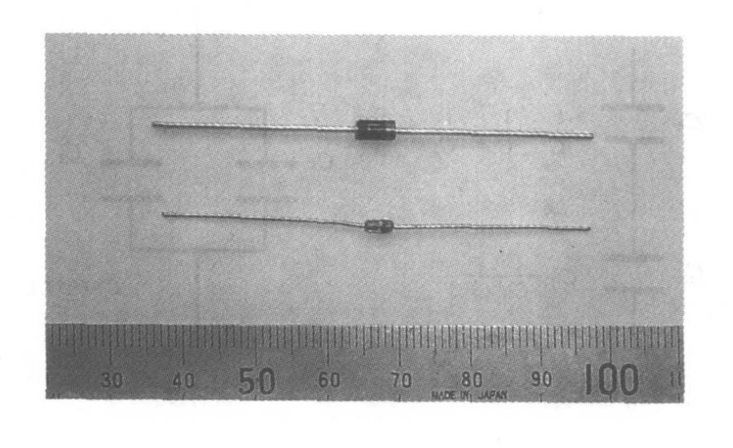
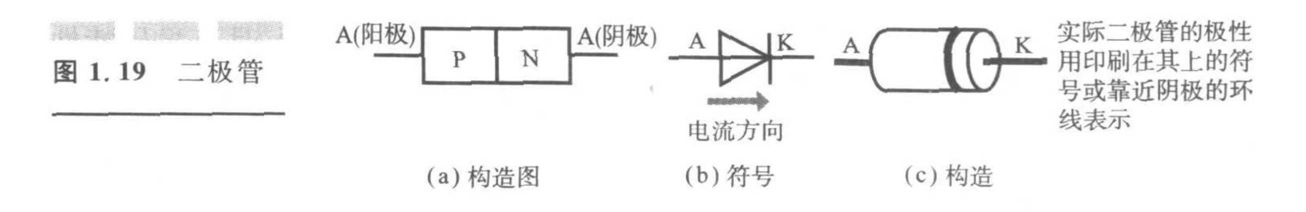


图 1.18　二极管的外形

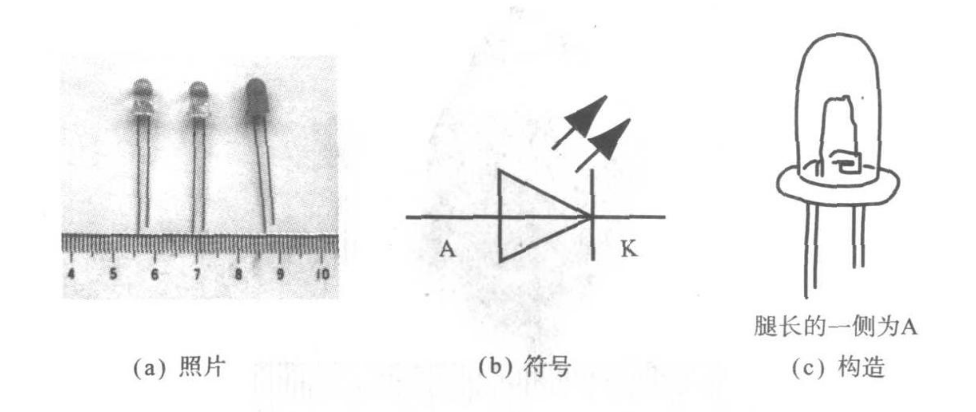
图 1.19 二极管



(a) 构造图　(b) 符号　(c) 构造

发光二极管(LED)的原理是在 PN 结的接合面上让电子与空穴再次结合,此时产生的能量以发光的形式释放出来。这并不是说所有的二极管都发光,只有直接跃变型半导体如钾-砷、铟、磷等能够发光,至于间接跃变型半导体如硅、锗等则不发光。图 1.20 为 LED 的外形及表示符号。

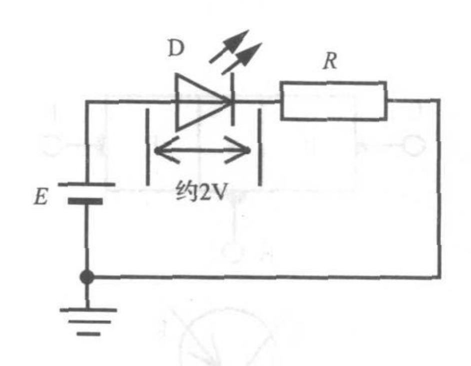
图 1.20 发光二极管



(a) 照片　(b) 符号　(c) 构造

近年来,随着对 LED 的进一步研究,具有各种各样发光颜色的 LED 被开发出来,发光的功率也越来越大。通过多个发光颜色的组合,可以显现更真实的颜色。如与白色接近的发光二极管可被用于照明。机器人电路中一般利用 LED 表示信号,或用作红外传感器中的发光元件。采用 LED 进行发光时,为了限制电流的大小,一般应将 LED 与电阻串联使用,如图 1.21 所示。

图 1.21 发光二极管的使用方法



$$R(\Omega)=\frac{\text{电源电压}(E)-\text{LED正电压}}{\text{LED的额定电流}}$$

例如:电源电压为5V,电流为10mA时

$$R(\Omega)=\frac{5-2\text{V}}{10\text{mA}}=300\Omega$$

在发红光时向 LED 施加的电压大约为 2V(电压的大小随 LED 种类的不同而异)。若由电源直接给 LED 施加 2V 以上的电压,则二极管将会有过大的电流流过,以至损坏二极管,所以需要通过串联一个电阻来限流。

还有利用反向特性的二极管，例如具有电容功能的**变容二极管**、具有稳压特性的**齐纳**(稳压)**二极管**及检测光信号的**光电二极管**等。此外，还包括具有负电阻特性的**隧道二极管**等。

1.4.5　晶体管

晶体管也是一种半导体，有放大信号的作用和对电流流动的**开关**(ON/OFF)**作用**。在机器人中主要是利用它的开关特性来控制电机。当然，计算机中的大脑，即 CPU(中央处理器)也是基于这种开关特性工作的。图 1.22 为晶体管的外形。

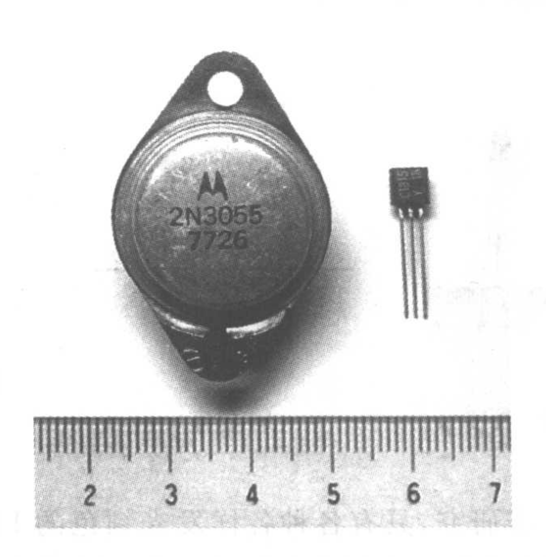


图 1.22　晶体管的外形

根据工作原理，晶体管可分为双极性型和单极性型。一般情况下，把**双极性晶体管**称为晶体管，而把**单极性晶体管**称为 FET(场效应晶体管)。晶体管根据其构造有 **NPN 型**和 **PNP 型**两种，两者中电流流动的方向相反。图 1.23 的表示符号中，指向发射极的箭头方向就是电流流动的方向。

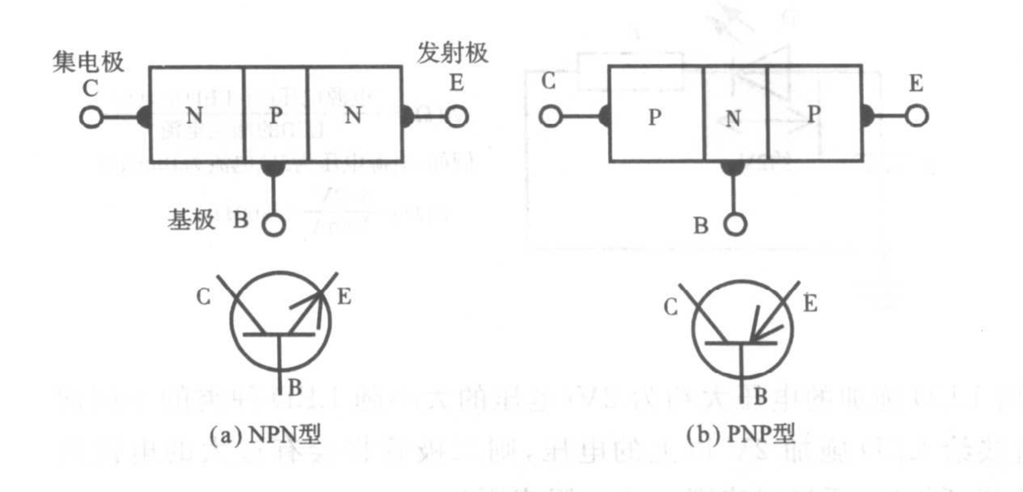
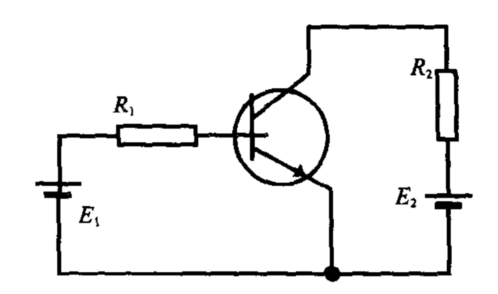


图 1.23　晶体管的构造及表示符号

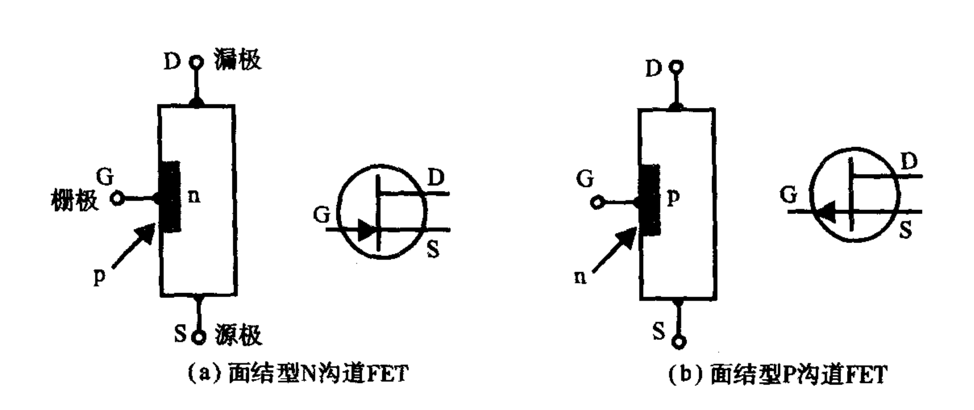
为了让晶体管工作，其基本电路应该如图 1.24 所示与两个电池相连接。但是实际上，总是设法使用一个电池来达到同样的目的。

图 1.24 晶体管的基本电路



FET 动作时，电流不流入栅极，漏极电流随着向栅极所施加电压的大小而变化。如果漏极电流来自于空穴的移动则称之为 P 沟道型，如果漏极电流来自于电子的移动则称之为 N 沟道型。沟道有如水道，就是电流的通道。图 1.25 是面结型 FET 的构造和表示符号。

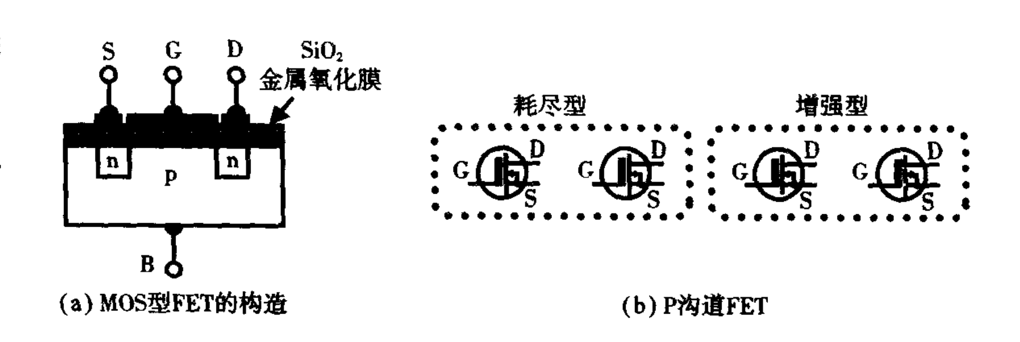
图 1.25 面结型 FET 与表示符号



(a) 面结型N沟道FET (b) 面结型P沟道FET

在面结型 FET 中，将 P 型半导体与 N 型半导体结合，在 PN 结合处施加反向电压即可进行栅极控制。此外，还有一种图 1.26 所示的用氧化膜对栅极起到绝缘作用的 MOS 型 FET（金属氧化物半导体场效应晶体管）。由于氧化膜非常薄、抗静电弱，所以在保管时应该用铝箔包装起来，最好插入导电泡沫中保存。

图 1.26 MOS 型 FET 的构造与符号



(a) MOS型FET的构造 (b) P沟道FET

根据工作特性，FET 可分为耗尽型和增强型，而兼有两者特性的 FET 叫做耗尽型＋增强型。

图 1.27 所示的耗尽型（含义为降低）FET 即使不施加栅极电压也有漏电流流过，所以，若向栅极施加反向电压，就将控制漏电流（I_D）减少的比例（面结型就是这种类型）。增强型（含义为提高）FET 当向栅极施加电压时，漏极与源极之间就会产生沟道，而有漏电流流过，于是，向栅极施加正向电压可以控制漏电流增加的比例（这种情形通常是 CMOS 型）。

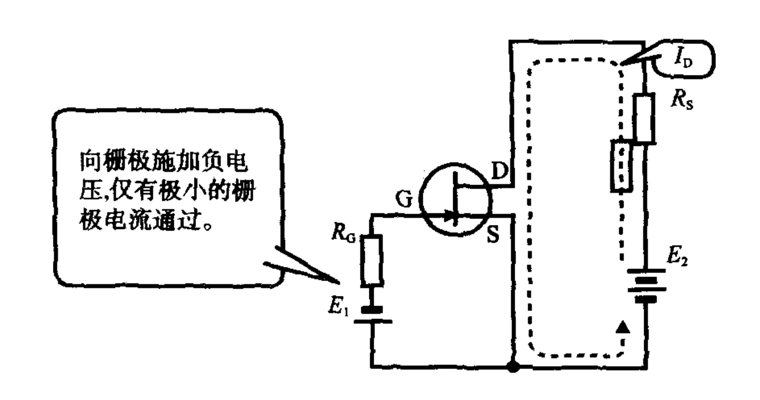


图 1.27　面结型 FET 的基本电路

1. 放大作用

晶体管的放大作用就是增大电流或电压。在图 1.28 中，若稍稍改变晶体管的基极电流，集电极电流 I_C 的变化将表现为该变化的 h_{fe}（晶体管的电流的放大倍数）倍。

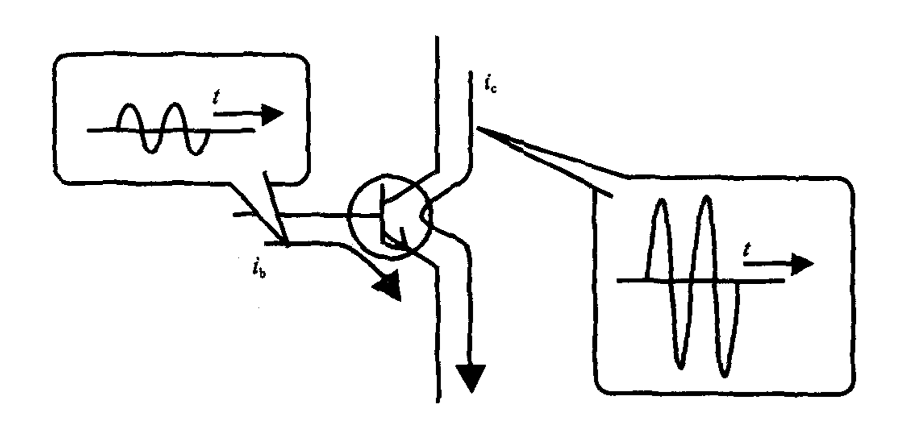


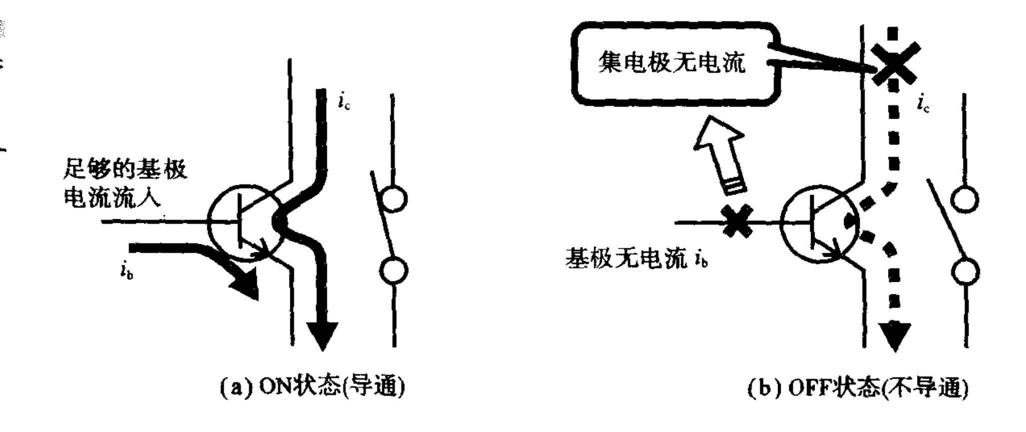
图 1.28　晶体管的增幅作用

2. 开关作用

在机器人中重点是利用晶体管的开关功能来控制电机的起停（ON/OFF）。开关只有 ON/OFF 动作，为在晶体管上实现这一功能是让集电极与发射集之间像开关一样导通或切断。具体地如图 1.29 所示，在让基极电流 i_b 充分流动达到饱和状态的条件下，集电极与发射集之间的电阻降低，于是晶体管处于 ON 的状态（导通）。反之，基极电流 i_b 为 0 时，在集电极与发射集之间电流不流动，晶体管处于 OFF 状态（不导

通)。若基极电流达到一半大小,那么晶体管的工作可能不稳定,这是要注意的地方。

图 1.29 晶体管的开关动作

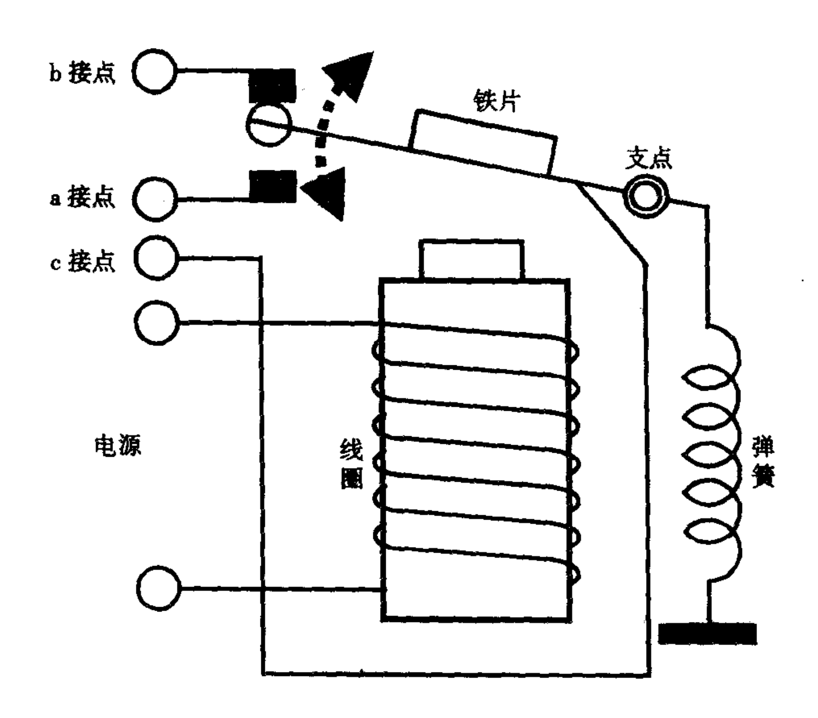


采用 FET 的开关电路与晶体管的原理相同,控制栅极电压即可以使漏电流达到饱和或为 0。最新的 MOS 型 FET 的开关电阻已经做得非常小了,因此开关动作造成的能耗极小。

1.4.6 电磁继电器

图 1.30 所示是电磁继电器利用电磁铁的开关工作。继电器的作用是将控制电路与负载断开,在大电流或高电压的负载时能够保护控制电路。如果继电器的接点是机械式的,则必须考虑关闭大电流时接点的损耗。基于半导体开关作用的继电器称为无接点继电器,优点就在于无接点上。

图 1.30 电磁继电器的构造



1.4.7 IC(集成电路)

利用微细加工工艺将晶体管、二极管、电阻等器件集成在半导体上，就做成半导体器件，如数字 IC(集成电路)和模拟 IC(集成电路)。

1. 数字 IC(集成电路)

针对数字 IC 内部的数字量的运算称为逻辑运算，其电路就是逻辑电路。基本的逻辑电路有 AND(与)电路，OR(或)电路，NOT(非)电路等。图 1.31 给出了符号、逻辑式、等效开关电路、输入/输出关系真值表等。

AND(逻辑与)电路

$Y=A \cdot B$

灯泡符号

输入		输出
A	B	Y
0	0	0
0	1	0
1	0	0
1	1	1

OR(逻辑或)电路

$Y=A+B$

输入		输出
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	1

NOT(逻辑非)电路

$Y=\overline{A}$

灯泡按钮，按下断开，离开则接通

输入	输出
A	Y
0	1
1	0

NAND(逻辑与非)电路

$Y=\overline{A \cdot B}$

输入		输出
A	B	Y
0	0	1
0	1	1
1	0	1
1	1	0

OR(逻辑或非)电路

$Y=\overline{A+B}$

输入		输出
A	B	Y
0	0	1
0	1	0
1	0	0
1	1	0

EXOR(排他逻辑和)电路

$Y=\overline{A} \cdot B+A \cdot \overline{B}$

输入		输出
A	B	Y
0	0	0
0	1	1
1	0	1
1	1	0

图 1.31　逻辑符号、逻辑式及真值表

无论 IC 内部电路是用晶体管构成的 TTL 型还是用 FET 构成的 CMOS 型，它们的逻辑符号相同。图 1.32 给出了几种实际的 IC 照片。

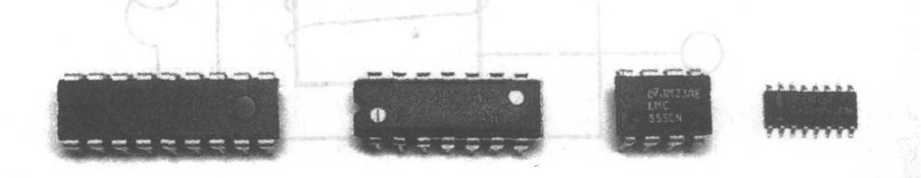


图 1.32　几种常用的 IC

根据制造厂家、制作方法及电路不同，数字 IC 的构成有表 1.4 所示的几种系列。各种系列有不同的特性，请根据不同的用途分别选用。

表 1.4 常用数字 IC 的种类

电路方式	各种系列	型　号	电源电压/V
TTL	TTL	74x	4.75～5.25
	LSTTL	74LSxx	
	STTL	74Sxx	
	FAST	74Fxx	4.5～5.5
	ALS TTL	74ALSxx	
	AS TTL	74ASxx	
COMS	标准 CMOS	40xx/45xx	3～18
	高速 CMOS	74HCxx	2.0～6.0
		74HCTxx	4.5～5.5
	高级 CMOS	74ACxx	2.0～6.0
		74ACTxx	4.5～5.5
BICMOS	BC	74BCxx	4.5～5.5
	BCT	74BCTxx	
	ABT	74ABTxx	

注：有一些特殊的 IC 如 ECL，但一般很少采用。

在型号 74xx 的后面还有后缀 xx，这是附加编号。如 NAND 电路，型号名为 7400。加后缀 74HCxx 即为 74HC00。需要提及的是，并非所有系列数字 IC 都具有全部逻辑电路型号产品。另外，虽然 MOS 型数字 IC 对电源电压的适应性较广，但也存在静电损坏的危险。所以选择时必须弄清楚各种 IC 的具体规格。

每一种 IC 的封装除以塑性为主外还有其他材质，形式也是多样的。不过制作机器人时，一般采用便于加工和订购的塑料 PID 封装 IC。举例来说，其中的 NOT 电路及 NAND 电路按图 1.33 所示的四个电路封装成 IC。从图 1.33 可知，不管哪一种 IC，第 14 个引脚一定是电源端子的＋极，第 7 个引脚接地为－极。

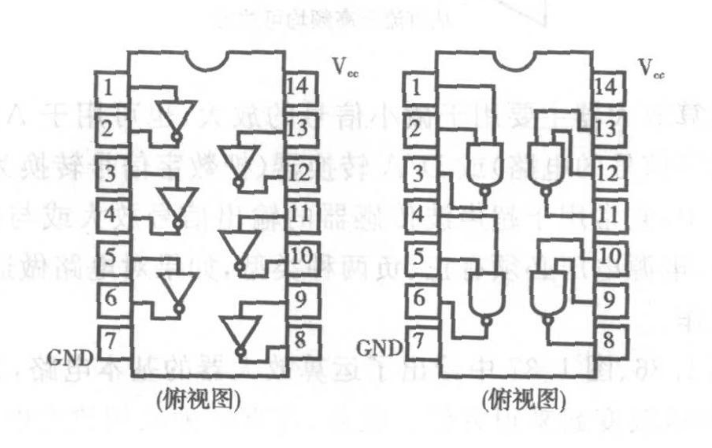


图 1.33 数字 IC 的引脚分布图

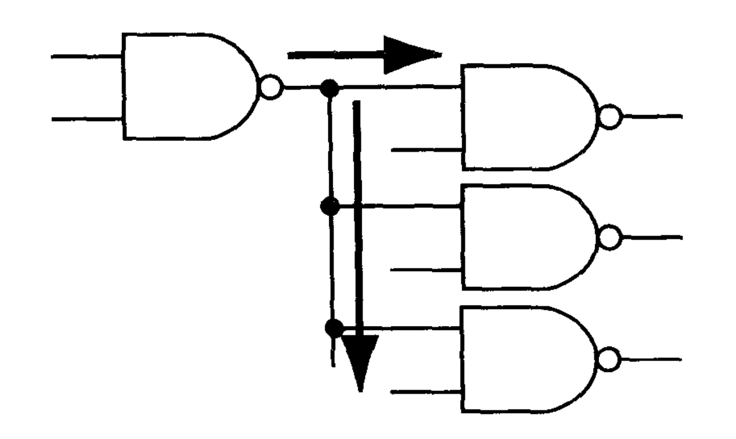


图 1.34　数字 IC 的组合

如图 1.34 所示，逻辑电路可以组合，此时应注意允许通过的电流大小有限制，所以 IC 相互连接的数目也是有限的。同种类 IC，一般最多有 10 个外接端口。

将 AND、OR 电路等组合后可以产生所谓组合电路。除此之外，还有一些具备触发等记忆功能的所谓顺序电路。

2. 模拟 IC

典型的模拟 IC 是运算放大器或 OP 放大器。运算放大器如图 1.35 所示，它的输入阻抗非常大，而输出阻抗非常小，放大系数无限大，在频率方面也几乎可以从直流到高频。从工作原理上讲，它的作用是将施加在两个输入端口上的电压差放大。

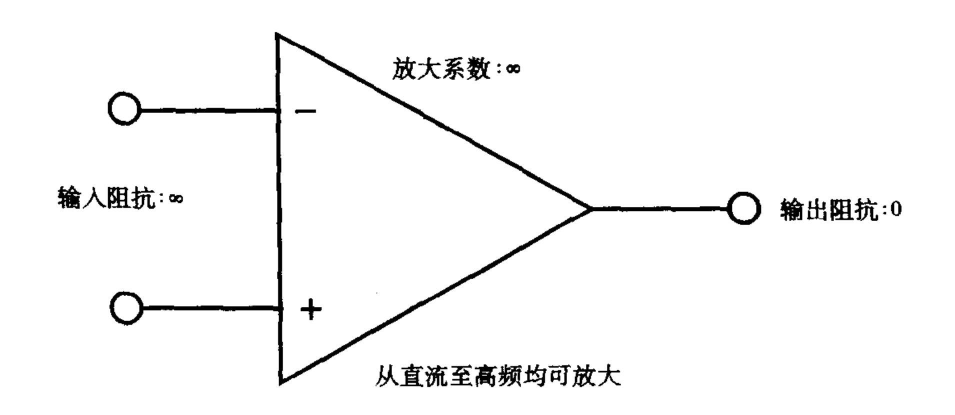


图 1.35　理想的运算放大器

运算放大器主要用于微小信号的放大，也可用于 A/D 转换器（把模拟信号转换为数字信号的电路）或 D/A 转换器（把数字信号转换为模拟信号的电路）等。在机器人中，它常用于超声波传感器的输出信号放大或与各种信号的比较。运算放大器中，电源电压必须有正、负两种类型，如果对电路做适当改进，也可以采用单个电源工作。

图 1.36、图 1.37 中给出了运算放大器的基本电路，其中放大系数 A 由电阻值确定，所以改变起来很方便。此外，还有一种反相放大电路，在这种电路中，输入信

号 V_i 与输出信号 V_o 的相位是倒相的(即相位正好相反)。

图 1.36　倒相放大电路

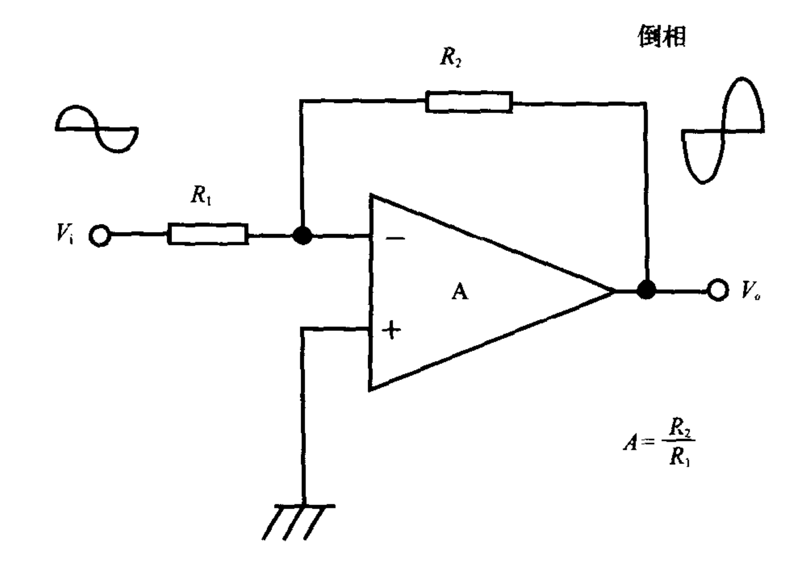
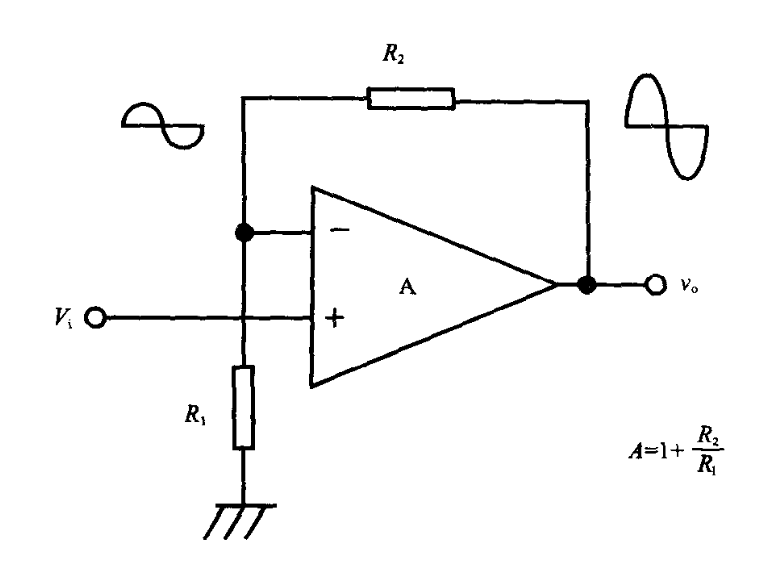


图 1.37　非倒相放大电路



练习题

1. 挑出下列装置中属于微控制器的组成部分,打上圆圈。

   输入　运算　存储　控制　外部存储　输出

2. 把下列二进制数或十六进制数表示为十进制数。

   ① $1010_{(2)}$　② $1110_{(2)}$　③ $111_{(2)}$　④ $F_{(16)}$　⑤ $A_{(16)}$　⑥ $D_{(16)}$

3. 求出下列各组电阻串联和并联后的合成电阻。

   ① $R_1=1\text{k}\Omega$,　$R_2=1\text{k}\Omega$

   ② $R_1=10\text{k}\Omega$,　$R_2=15\text{k}\Omega$

4. 根据下列电阻的彩色条码求电阻值的范围。

   ① 茶　黑　茶　银　　　　② 橙　橙　茶　金

5. 求下列各电容串联和并联后的合成电容。

   ① $C_1=1\mu F$，　$C_2=1\mu F$

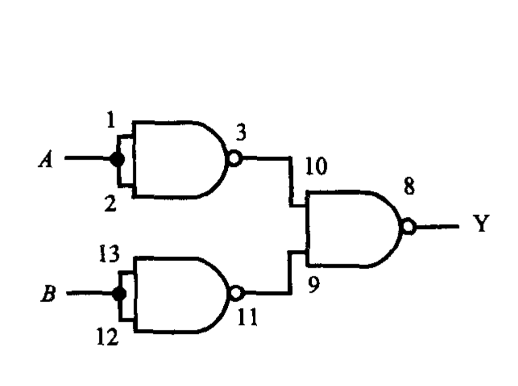
   ② $C_1=10\mu F$，　$C_2=15\mu F$

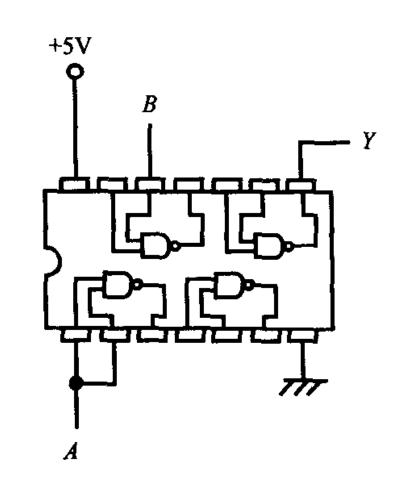
6. 欲设计由 4 个干电池串联的电源和发光 LED 构成的闪烁电路。如果电流 1mA，计算该电流下的电阻值，使用的电阻值从 E24 系列中选取。

7. 说明下列型号半导体的种类及用途。

   ① 2SC1815　　　② 2SD1237　　　③ 2SA1972

8. 要使左边的图成为逻辑电路，请将右图的布线配齐（电源已如图配置完毕）。





9. 简单说明运算放大器的特点。

第2章

微控制器及编程

为了对机器人进行控制，经常采用微控制器。例如自主型机器人，需要先对所处周边环境状况作判断，然后采取相应的动作，所以通常都安装微控制器，它们就是机器人大脑。另外，在遥控操纵型机器人中，在遥控电路中采用微控制器的事例也屡见不鲜。

本章以性能价格比优良，且易于购入的 PIC 微控制器为例，简要地说明它的基本组成和使用方法。另外，对 H8 微控制器的组成及其概要也做了简单介绍。在学习过程中，读者也许会感到有点难度，但实际上本书并不涉及难度太大的内容。重要的是，一旦掌握这些基本知识就可以应用于很多领域。现在，就让我们从第一步开始学习吧！

2.1 微控制器的基础

机器人上的微控制器，其担当的任务多半是从各种传感器获取信号，对应于机器人和环境的状况，做出相应的动作反应。由此可知，微控制器通常扮演机器人大脑的角色。

在这里，先以在机器人比赛中颇为活跃的自主型相扑机器人为例来说明微控制器的功能。图 2.1 为自主型相扑机器人的外观，图 2.2 中给出位于机器人上部的电路部分，图 2.2 中央能看到一款称为 **Z80** 的微控制器。

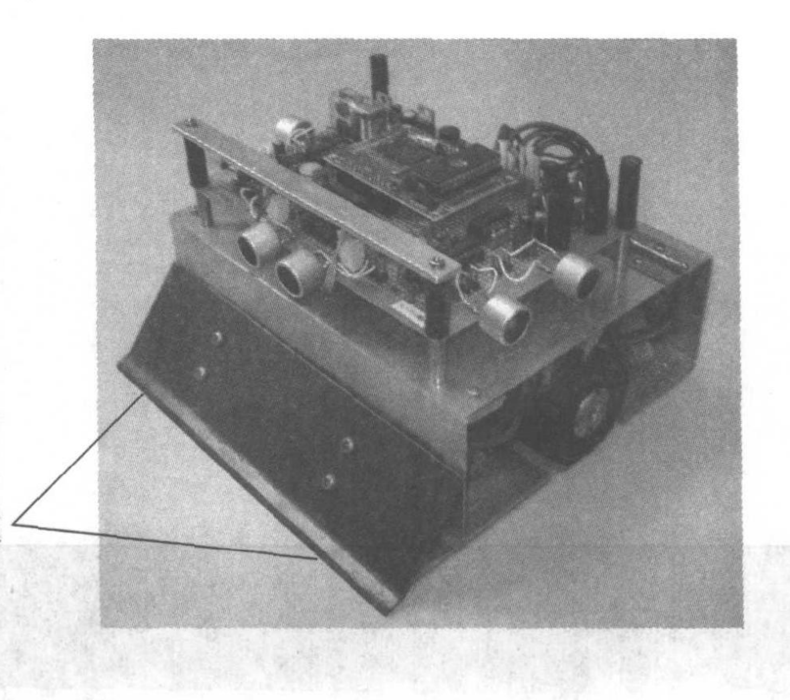
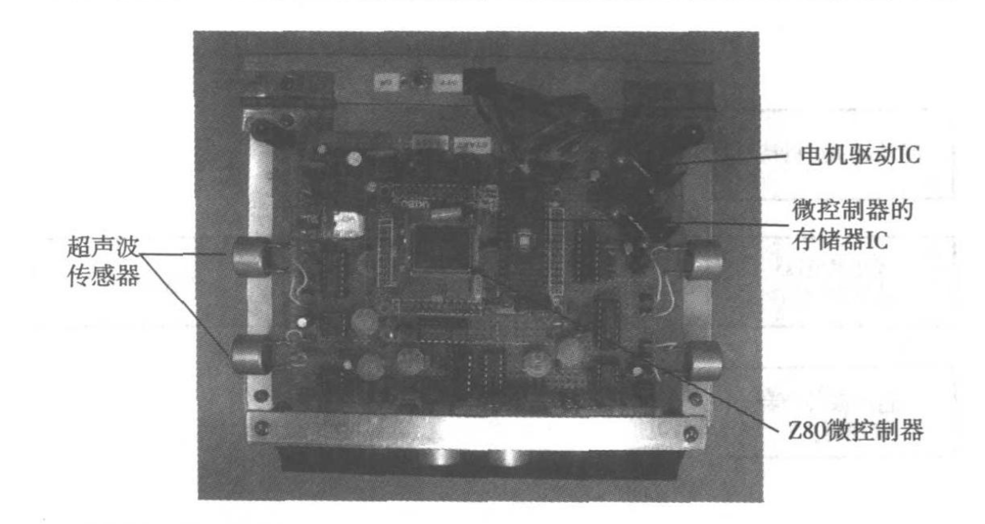


图 2.1　自主型相扑机器人的外观

图 2.2　电路部分



相扑机器人比赛是在直径为 154cm 的摔跤场上由两台机器人进行的角力对抗战。比赛的规则是把对方机器人从摔跤场推出界就得 1 分，先取得 2 分者胜出。这个比赛有自主型和无线遥控型两种。在裁判发出比赛开始的信号后，先由参赛队员按下起动开关，此后机器人就完全自主地行动，不再允许外界对机器人发出任何命令。按照规则，机器人必须在按下起动开关后 5 秒钟才能动作(参见图 2.3)。

图 2.3　自主型机器人相扑的比赛规则

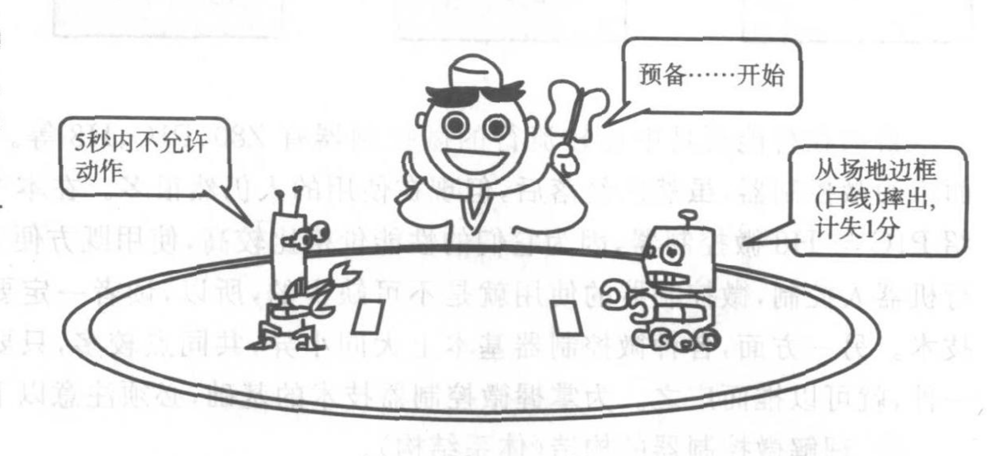


图 2.1 所示的自主型相扑机器人具有以下功能：

① 按下起动开关 5 秒钟后开始动作。

② 正面的下方在左右 2 处分别安装光电传感器以感知摔跤场的边框，这样机器人就能够躲开边框而不会掉下土台。

③ 机器人上部在前端及左右 3 处分别安装超声波传感器，以寻找对方机器人，然后朝着对方前行。

④ 按照比赛前所设定的模式开始动作。

以上这些动作均由机器人上搭载的微控制器控制，图 2.4 中给出了这类机器人的构成。

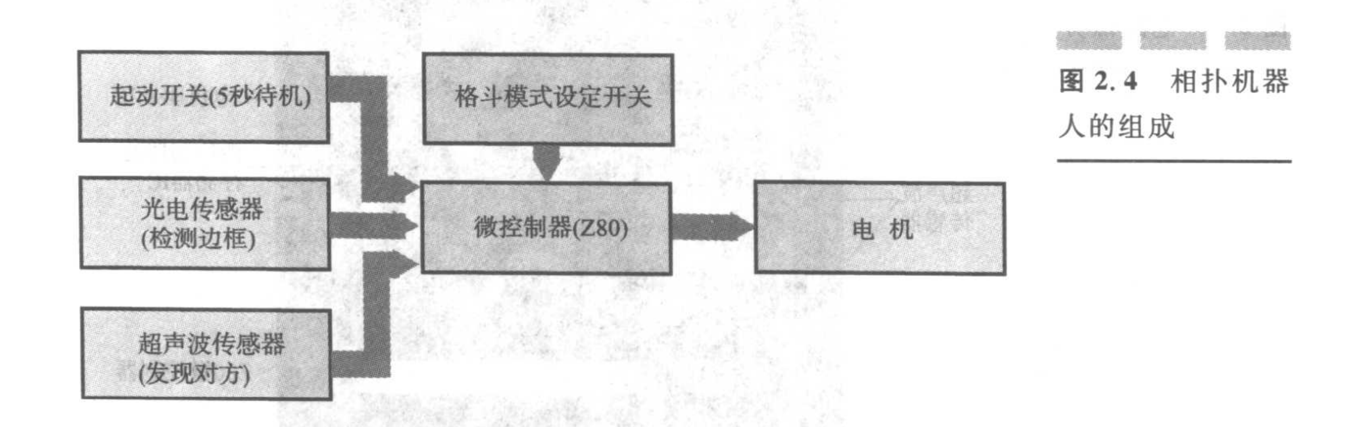


图 2.4 相扑机器人的组成

从图 2.4 可知，由于机器人上搭载了微控制器，所以能根据环境状况自主地进行动作。虽然微控制器的种类很多，但基本处理流程都是相同的，也就是说，把来自传感器的信号作为输入，经过必要的加工处理，向电机发出信号(参见图 2.5)。

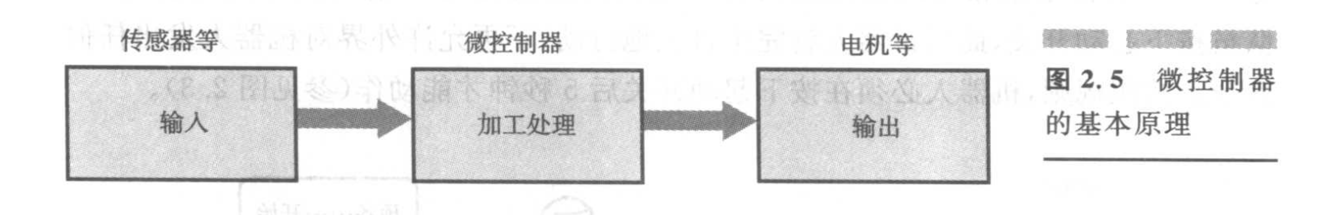


图 2.5 微控制器的基本原理

目前在智能玩具中比较流行的微控制器有 Z80、P1C、H8 等。Z80 是 1976 年面市的微控制器，虽然已经落后，但现在使用的人仍然很多。在本书中打算简要介绍 P1C 与 H8 微控制器，因为它们的性能价格比较高，使用既方便又简单。既要进行机器人控制，微控制器的使用就是不可缺少的，所以，读者一定要掌握微控制器技术。另一方面，各种微控制器基本上大同小异，共同点较多，只要掌握其中任何一种，就可以推而广之。为掌握微控制器技术的基础，必须注意以下三个方面：

① 理解微控制器的构造(体系结构)。

② 理解相关的编程知识(**汇编语言或 C 语言**)。

③ 尝试简单的实际控制。

2.2 PIC 微控制器

PIC 是外围设备接口控制器的简称。在微控制器中将之昵称为“**猪**”。本节来学习它的结构组成和使用方法。

2.2.1 什么是 PIC

PIC 是美国 Microchip Technology 公司开发的产品，在日本也有它的分公司（Microchip Technology Japan）。目前，PIC 已经被广泛地搭载在各种电气制品和汽车，不露声色地发挥着重要的作用。由于 PIC 的使用很简单，近年来，用于电子产品制作的 PIC 也逐渐增多。图 2.6 给出了各种 PIC 的照片。

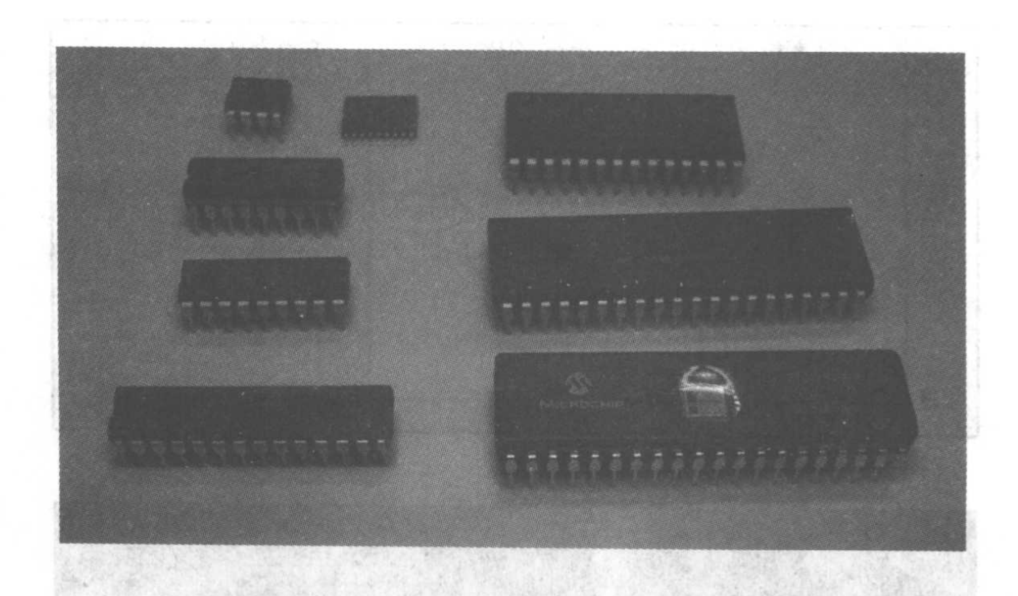


图 2.6 各种 PIC 实物照片

PIC 是一种集 CPU（中央处理器）、RAM、ROM、输入/输出端口、定时器、时钟振荡电路为一体的高性能微控制器，封装得十分紧凑。人们往往把这种一个大规模集成电路 LSI 上集成了多种功能的微处理器称为单片机，如图 2.7 所示。

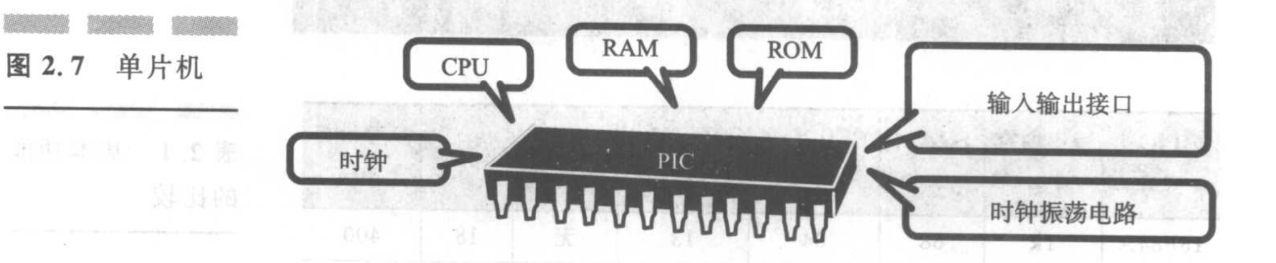


图 2.7 单片机

2.2.2 PIC 的种类

如图 2.8 所示，PIC 可分为基本产品、中档产品和高档产品三大类。

基本 PIC 产品中，命令的字长（构成一个命令的位长度）为 12 位，因此它的价格比较低，常用的型号有 **12C508、16C54** 等。

中档产品是指命令字长以 14 位为主的 **PIC 微控制器**。常见的型号有 **16F84A、16F877A**。

高档产品的命令字长达到 16 个位，属于高性能微控制器，常见的型号有

17C44、18F452 等。

在应用场合，若控制方式简单，就可以选用 16F84A，若需要较高的性能，就可以选用 16F877A。这两种 PIC 属于中档产品。图 2.9 中给出了 16F84A 和 16F877A 的外形。表 2.1 给出它们基本功能的比较，可以看出，16F84A 的容量大，而且内部还藏有 A/D 转换器。

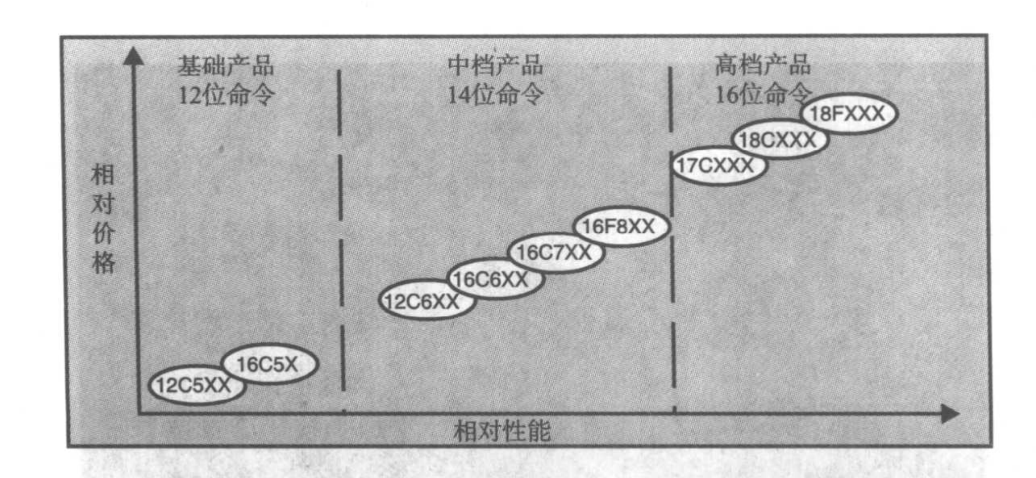


图 2.8　PIC 的分类（摘自 Microchip Technology 产品目录）

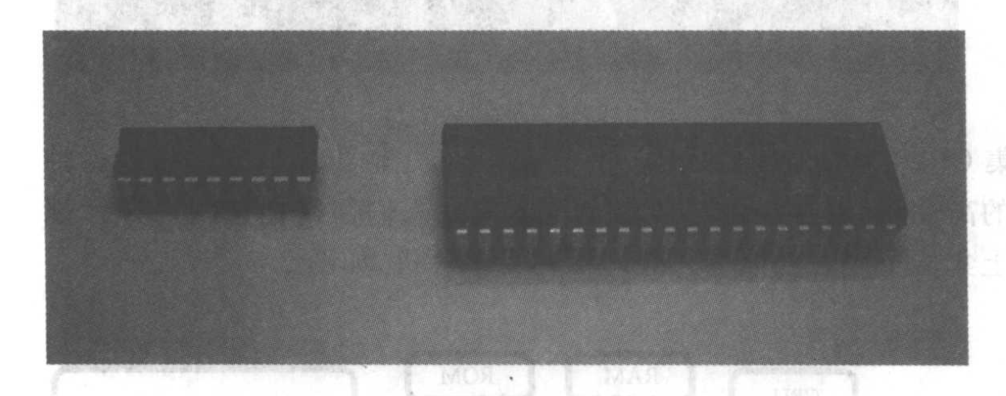


图 2.9　16F84A（左）和 16F877A（右）的外形

表 2.1　基本功能的比较

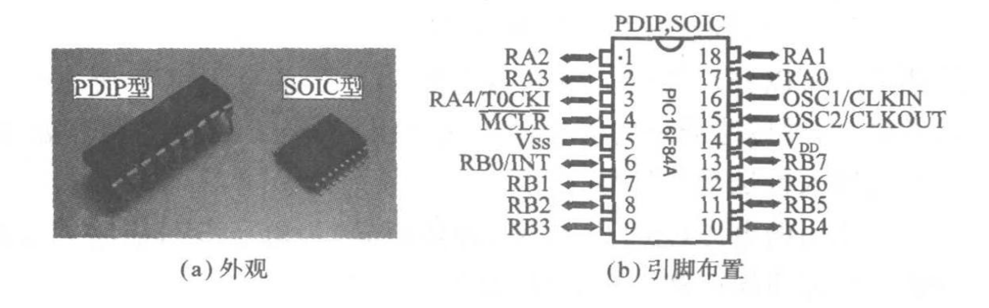
PIC 型号	程序内存 /字	SRAM /字节	EEPROM /字节	（输入/输出端口）/个	A/D 转换器	引脚数	参考价格 /日元
16F84A	1k	68	64	13	无	18	400
16F877A	8k	368	256	33	内置	40	800

2.2.3 PIC 的组成

下面以 16F84A 为例讲解 PIC 的基本组成。

16F84A 是一个小型微控制器，有 18 条引脚，作为玩具控制用途，它的功能应该是足够的。图 2.10(a)中为 PDIP 型与 SOIC 型的外观，图 2.10(b)为引脚配置图。表 2.2 给出各引脚定义说明。

图 2.10　16F84A



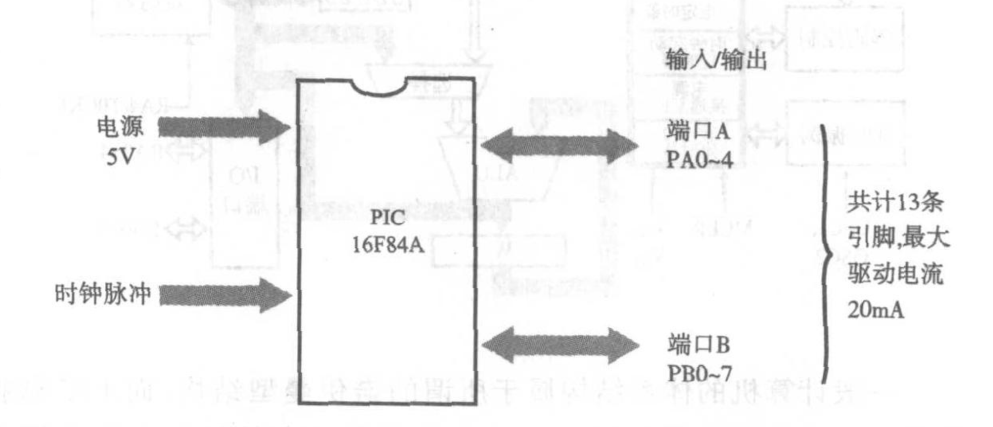
(a) 外观　　(b) 引脚布置

表 2.1　F84A 引脚的定义

引脚编号	名　称	说　明
1	RA2	双向 I/O 端口(端口 A 的第 2 位)
2	RA3	双向 I/O 端口(端口 A 的第 3 位)
3	RA4/T0CKI	双向 I/O 端口(端口 A 的第 4 位) 定时器时钟输入
4	$\overline{\text{MCLR}}$	复位
5	Vss	接地(接电源的负接线端子)
6	RB0/INT	双向 I/O 端口(端口 B 的第 0 位),外部中断
7	RB1	双向 I/O 端口(端口 B 的第 1 位)
8	RB2	双向 I/O 端口(端口 B 的第 2 位)
9	RB3	双向 I/O 端口(端口 B 的第 3 位)
10	RB4	双向 I/O 端口(端口 B 的第 4 位)
11	RB5	双向 I/O 端口(端口 B 的第 5 位)
12	RB6	双向 I/O 端口(端口 B 的第 6 位)
13	RB7	双向 I/O 端口(端口 B 的第 7 位)
14	V_{DD}	电源正极接线端子
15	OSC2/CLKOUT	时钟端口 2
16	OSC1/CLKIN	时钟端口 1
17	RA0	双向 I/O 端口(端口 A 的第 0 位)
18	RA1	双向 I/O 端口(端口 A 的第 1 位)

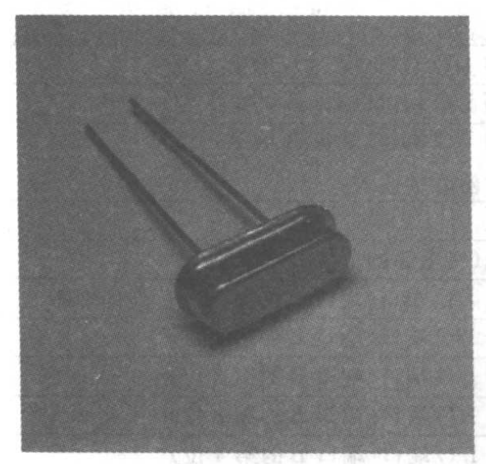
为使 16F84A 正常运行,应该按照图 2.11 所示完成连接。

图 2.11　16F84A 的连接



16F84A微控制器的输入/输出引脚共计有13条,其中端口A 5条、端口B 8条。这些引脚可以分别设定为输入或者输出。一条引脚可驱动的最大电流为20mA。或者说,各种传感器来的信号输入这些引脚,经PIC微控制器处理后,再从输出引脚向机器人传送动作信号。

工作用的电源(随工作频率、种类而异)为电压5V,应将它连接到产生工作时钟的石英振荡器或**陶瓷振荡器**(参见图2.12)。



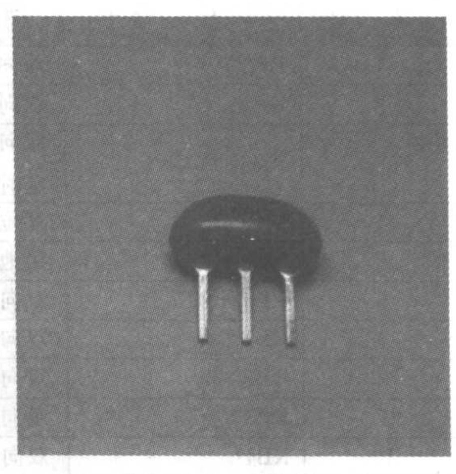


图2.12 石英振荡器(左)与陶瓷振荡器(右)

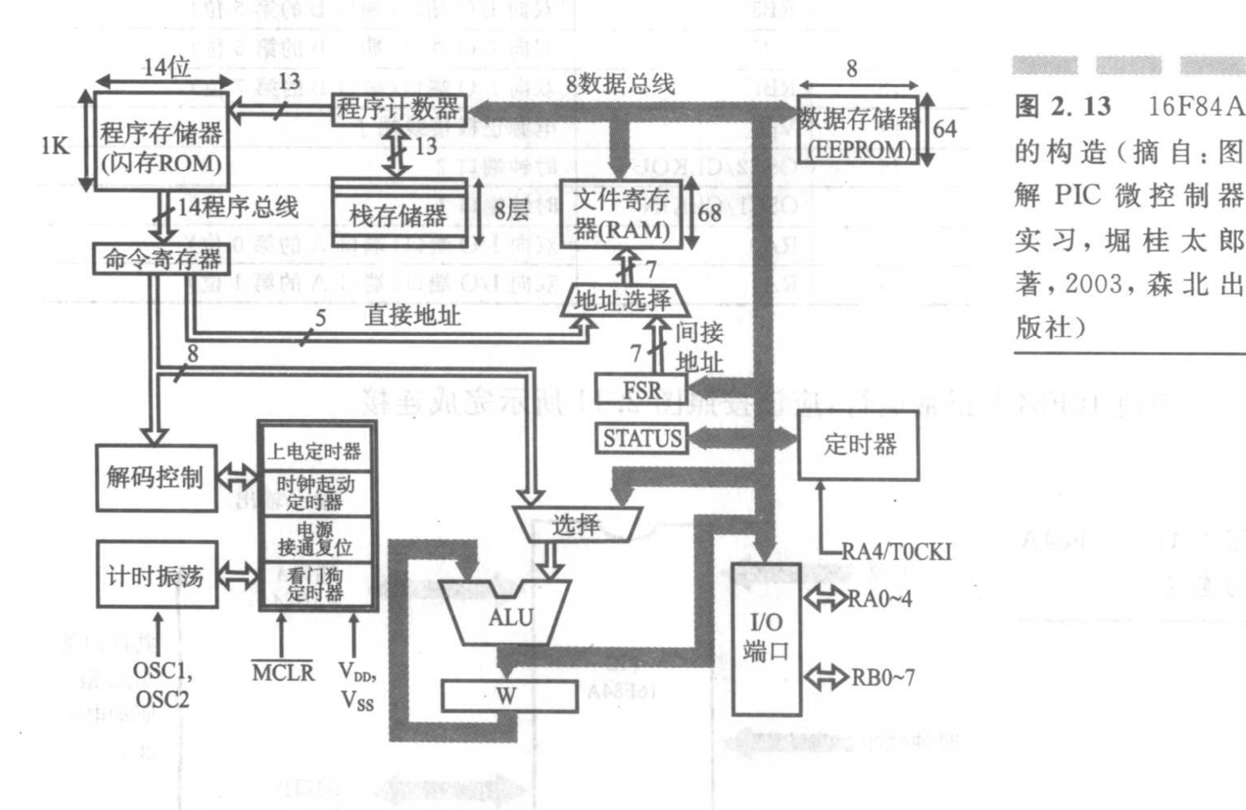


图2.13 16F84A的构造(摘自:图解PIC微控制器实习,堀桂太郎著,2003,森北出版社)

一般计算机的体系结构属于所谓的**诺伊曼型**结构,而PIC微控制器则是**哈佛型**结构。哈佛型结构的特征在于存储程序的存储器与存储数据的存储器是分离

的，这样，虽然结构比诺伊曼型的复杂，但便于高速处理。图2.13给出了16F84A的结构组成。其中，ALU表示算术逻辑运算器，W表示暂存寄存器，它们对程序的运行都负有重要的任务。

16F84A中包含表2.3所示的4种存储器。

表2.3 16F84A的存储器

存储器	功能	数据长度/位	容量	类型	写入	电源OFF时的数据状态
程序	存储程序	14	1024(1K)	闪存ROM	ROM写入装置	保持
文件寄存器	由SFR与通用寄存器组成	8	68	RAM	程序命令	消失
数据	存储数据	8	64	EEPROM	程序命令(经SFR)	保持
栈	存储子程序的返回地址	13	8	RAM	程序命令(CALL)	消失

存储程序的存储器是具有1K字容量的**闪存只读存储器**。PIC所有的每一个命令都为一个字长，所以16F84A可存储1024条命令。而且PIC内藏的闪存ROM是一种电可擦可写的IC模块。

下面来看看PIC输入/输出引脚的使用方法。使用I/O引脚时，将进行下列操作：

① 设置端口的引脚为输入端口或输出端口。

② 设置从端口输入引脚读取输入数据，或向端口输出引脚传送输出数据。

例如，考虑若设置端口B的上4位为输入引脚，下4位为输出引脚。端口B的输入/输出设置靠叫做TRISB的8位寄存器完成。寄存器TRISB的各位与端口B的各位相对应，比如，若把“0”写入TRISB，端口B的对应位就被设置为输出引脚，反之，若把“1”写入TRISB，端口B的对应位就被设置为输入管脚(参见图2.14)。寄存器TRISB位于文件寄存器的RAM存储单元第1列的86号地址，见图2.15。

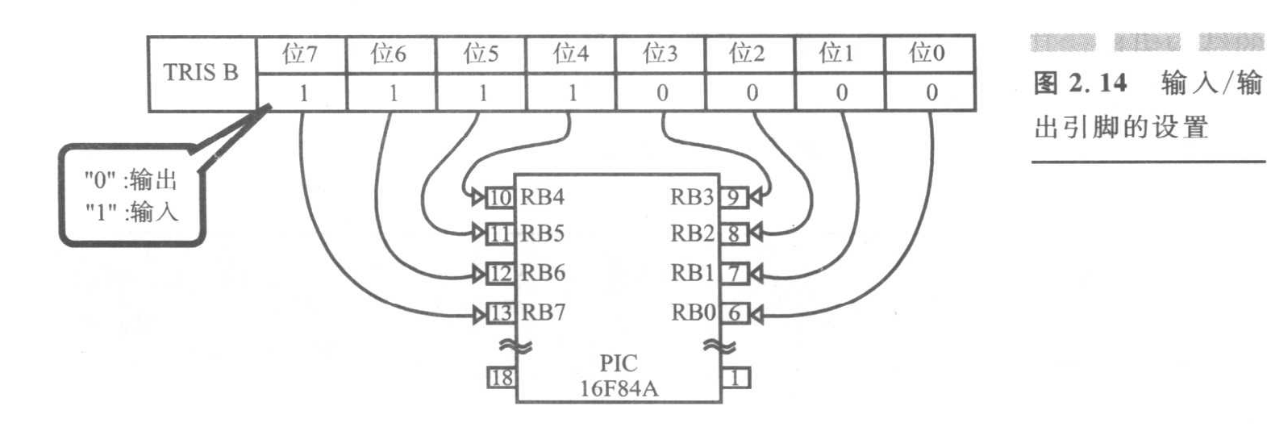


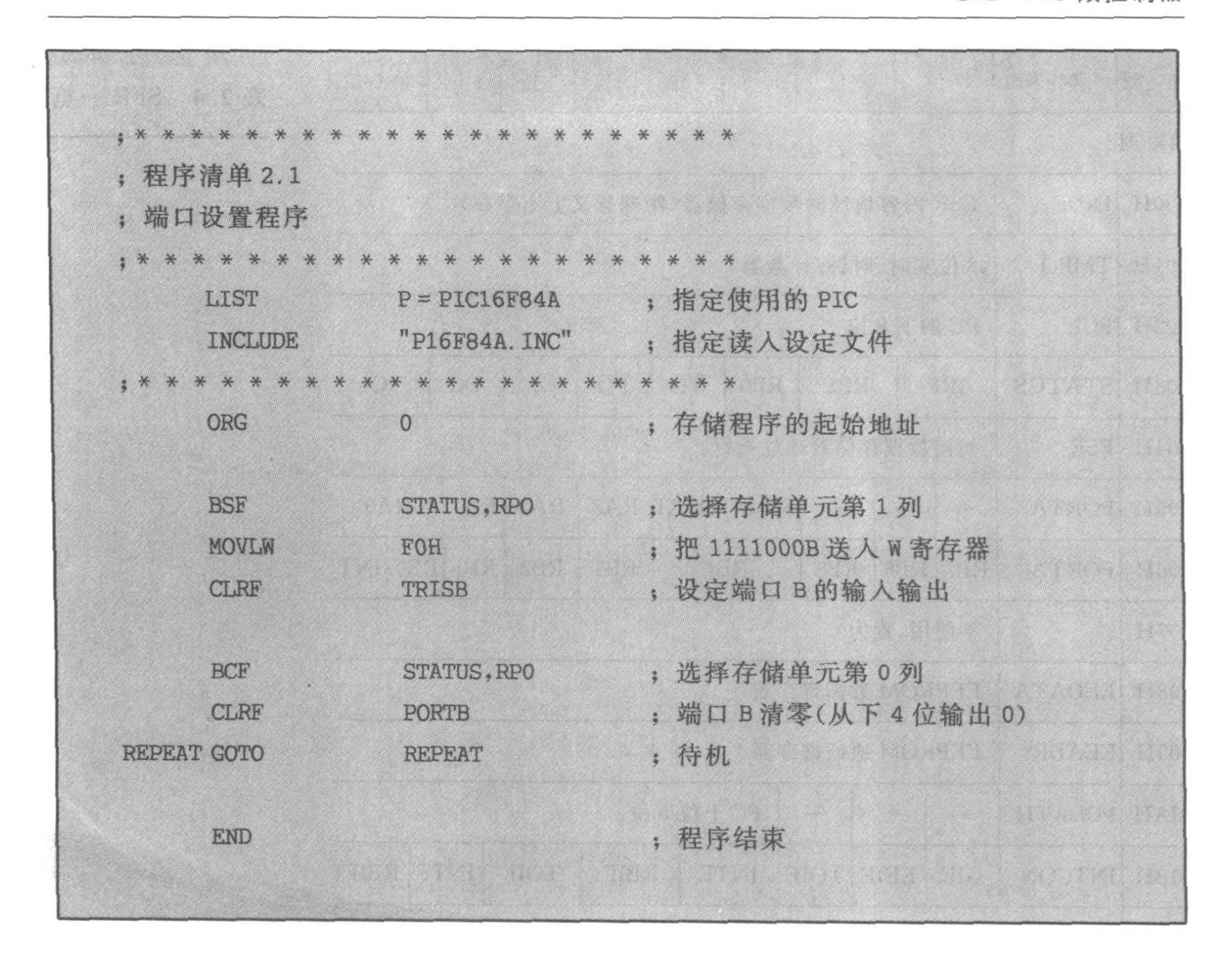
图 2.14 输入/输出引脚的设置

地址	存储单元第0列	存储单元第1列	地址
00	INDF	INDF	80
01	TMR0	OPTION_REG	81
02	PCL	PCL	82
03	STATUS	STATUS	83
04	FSR	FSR	84
05	PORTA	TRISA	85
06	PORTB	TRISB	86
07	未使用	未使用	87
08	EEDATA	EECON1	88
09	EEADR	EECON2	89
0A	PCLATH	PCLATH	8A
0B	INTCON	INTCON	8B
0C–4F	通用寄存器 68个	与第0列相同的通用寄存器	8C–CF

（地址00～0B为SFR）

图 2.15 文件寄存器

这些设置若用 PIC 汇编语言编写，就是程序清单 2.1。



```
;* * * * * * * * * * * * * * * * * * * * * * * *
; 程序清单 2.1
; 端口设置程序
;* * * * * * * * * * * * * * * * * * * * * * * *
       LIST        P = PIC16F84A       ; 指定使用的 PIC
       INCLUDE     "P16F84A.INC"       ; 指定读入设定文件
;* * * * * * * * * * * * * * * * * * * * * * * *
       ORG         0                   ; 存储程序的起始地址

       BSF         STATUS,RPO          ; 选择存储单元第 1 列
       MOVLW       FOH                 ; 把 1111000B 送入 W 寄存器
       CLRF        TRISB               ; 设定端口 B 的输入输出

       BCF         STATUS,RPO          ; 选择存储单元第 0 列
       CLRF        PORTB               ; 端口 B 清零(从下 4 位输出 0)
REPEAT GOTO        REPEAT              ; 待机

       END                             ; 程序结束
```

程序清单 2.1 端口设置程序实例

想使用端口 A 时，与端口 B 的设置步骤相同，不过应该使用寄存器 TRISA 与 PORTA 的下 5 位。

使用 PIC 时，对图 2.15 所示的文件寄存器的上部(地址 00～0B、80～8B)中的 SFR(特殊功能的寄存器)的操作具有重要的意义。表 2.4 中给出了 SFR 的一览表，在 SFR 中地址 0C～4F 可以作为 RAM 区域使用。

PIC 提供直接操作 16F84A 的汇编语言命令总共只有 35 条，如表 2.5 所示。

由于 PIC 是一种通过少数命令的组合进行程序处理的即所谓 RISC 方式的微控制器，所以掌握汇编语言比较容易。除汇编语言外，后面也将说明如何采用 C 语言进行编程。

想学习关于 PIC 更详细的构造及编程方法的读者，请参阅本人的拙著《图解 PIC 微控制器实习》(森北出版社，2003)。

表 2.4　SFR 一览

地　址	名　称	7 位	6 位	5 位	4 位	3 位	2 位	1 位	第 0 位
第 0 列									
00H	INDF	ESR 内容地址的数据存储器(物理意义上不存在)							
01H	TMPO	8 位实时、时钟/计数器							
02H	PCL	PC 的下 8 位							
03H	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
04H	FSR	间接数据存储器地址指针							
05H	PORTA	—	—	—	RA4/TOCKI	RA3	RA2	RA1	RA0
06H	PORTA	RB7	RB6	RB5	RB4	RB3	RB2	RB1	RB0/INT
07H		不使用,置为 0							
08H	EEDATA	EEPROM 数据寄存器							
09H	EEADR	EEPROM 地址寄存器							
0AH	PCLATH	—	—	—	PC 上位 5 位				
0BH	INTCON	GIE	EEIE	TOIE	INTE	RBIE	TOIF	INTF	RBIF
第 1 列									
80H	INDF	FSR 内容地址的数据存储器(物理意义上不存在)							
81H	OPTION_REG	$\overline{RBPU}$	INTEDG	TOCS	TOSE	PSA	PS2	PS1	PS0
82H	PCL	PC 下 8 位							
83H	STATUS	IRP	RP1	RP0	$\overline{TO}$	$\overline{PD}$	Z	DC	C
84H	FSR	间接数据存储地址指针							
85H	TRISTA	—	—	—	PORTA 数据输入输出设定寄存器				
86H	TRISB	PORTB 数据输入/输出设置寄存器							
87H		不使用,置为 0							
88H	EECON1	—	—	—	EEIF	WRERR	WREN	WR	RD
89H	EECON2	EEPROM 控制寄存器 2(无物理内存)							
8AH	PCLATH	—	—	—	向 PC 上 5 位写入的缓冲器				
8BH	INTCON	GIE	EEIE	TOIE	INTE	PBIE	TOIF	INTF	RBIF

表 2.5 16F84A 的命令

命令的种类		命 令
字节操作命令（以文件寄存器的全部内容为对象）	传送	MOVF，MOVWF，SWAPF
	算术运算	ADDWF，SUBWF，INCF，DECF
	逻辑运算	ANDWF，IORWF，COMF，XORWF
	循环移位	RRF，RLF
	转移	INCFSZ，CLRWSZ
	清除	CLRF，CLRW
	特殊	NOP
位操作命令（以文件寄存器的指定位为对象）	位操作	BCF，BSF
	转移	BTFSC，BTFSS
文字（数据）操作命令与控制命令	传送	MOVLW
	算术运算	ADDLW，SUBLW
	逻辑运算	ANDLW，IORLW，XORLW
	转移	CALL，GOTO，RETURN，RETLW，RETFIE
	清除	CLRWDT
	控制	SLEEP

（摘自：图解 PIC 微控制器实习，森北出版社，2003）

2.2.4 PIC 程序开发环境

使用 PIC 开发程序时按如图 2.16 所示的流程进行。

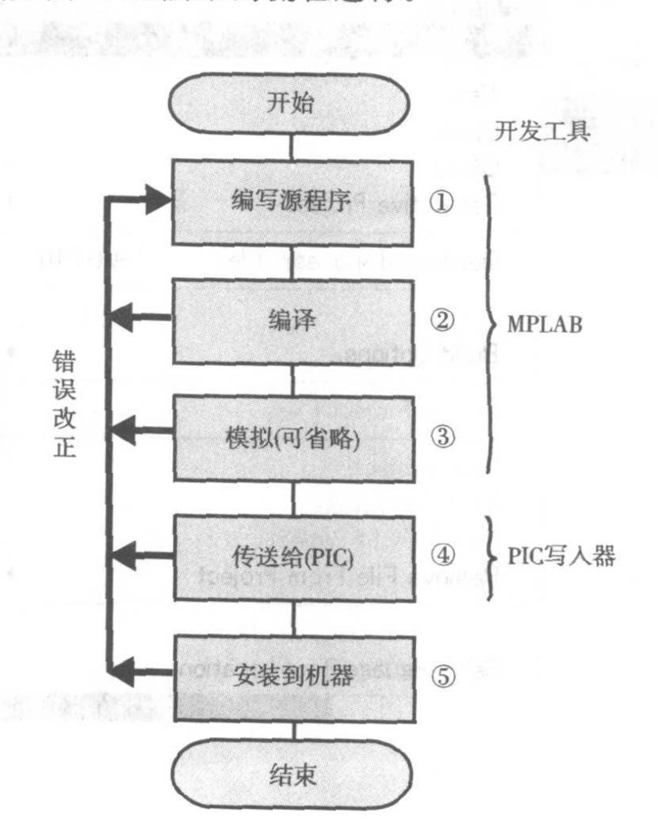


图 2.16 PIC 程序开发流程

1. PIC 程序开发流程

(1) **编写源程序**　根据汇编语言编写源程序。

(2) **编译**　把源程序转换成机器语言。

(3) **模拟**　把编好的程序在计算机上通过相关软件模拟执行过程。如果程序很简单也可以省略此步骤。

(4) **传送给 PIC**　借助 **PIC 写入器**(例如 PICSTART plus 等),把编译完成的机器语言程序传送给 PIC 中的 ROM。

(5) **安装 PIC**　把完成程序传送的 PIC 装入机器人本体上,确认最后的动作。

图 2.17 给出了开发程序所需要的工具

上面的步骤(1)～(3)在开发软件 MPLAB 环境下进行,而 MPLAB 应该由 PIC 开发商 Microchip technology 公司免费提供。如果上面第(4)步骤的 PIC 写入器使用 Microchip 公司供应的 **PICSTART plus**,则价格在 3 万日元左右。如果改成秋月电子通商(http://akizakidenshi.com/)编程器,价格将会更便宜。图 2.18 是 PICSTART plus 的外形照片。图 2.19 是秋月电子通商 **AKI-PIC 编程器**的外形照片。

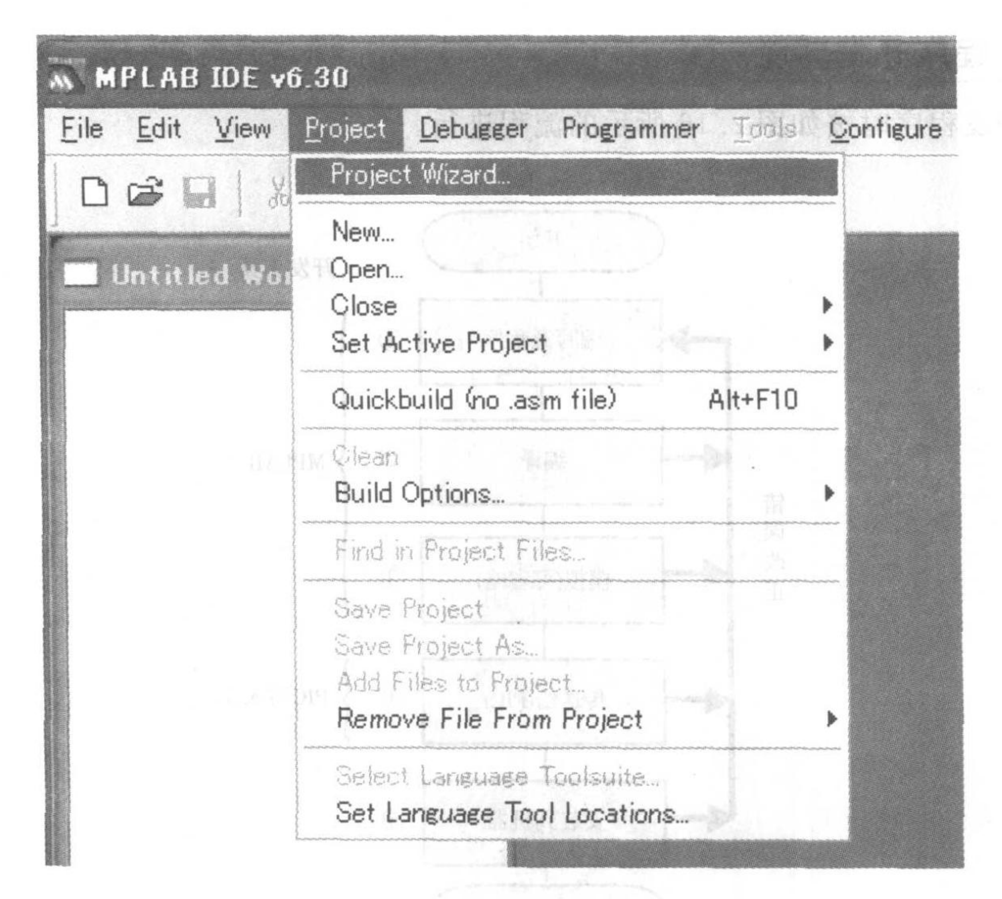


图 2.17　开发所需要的工具

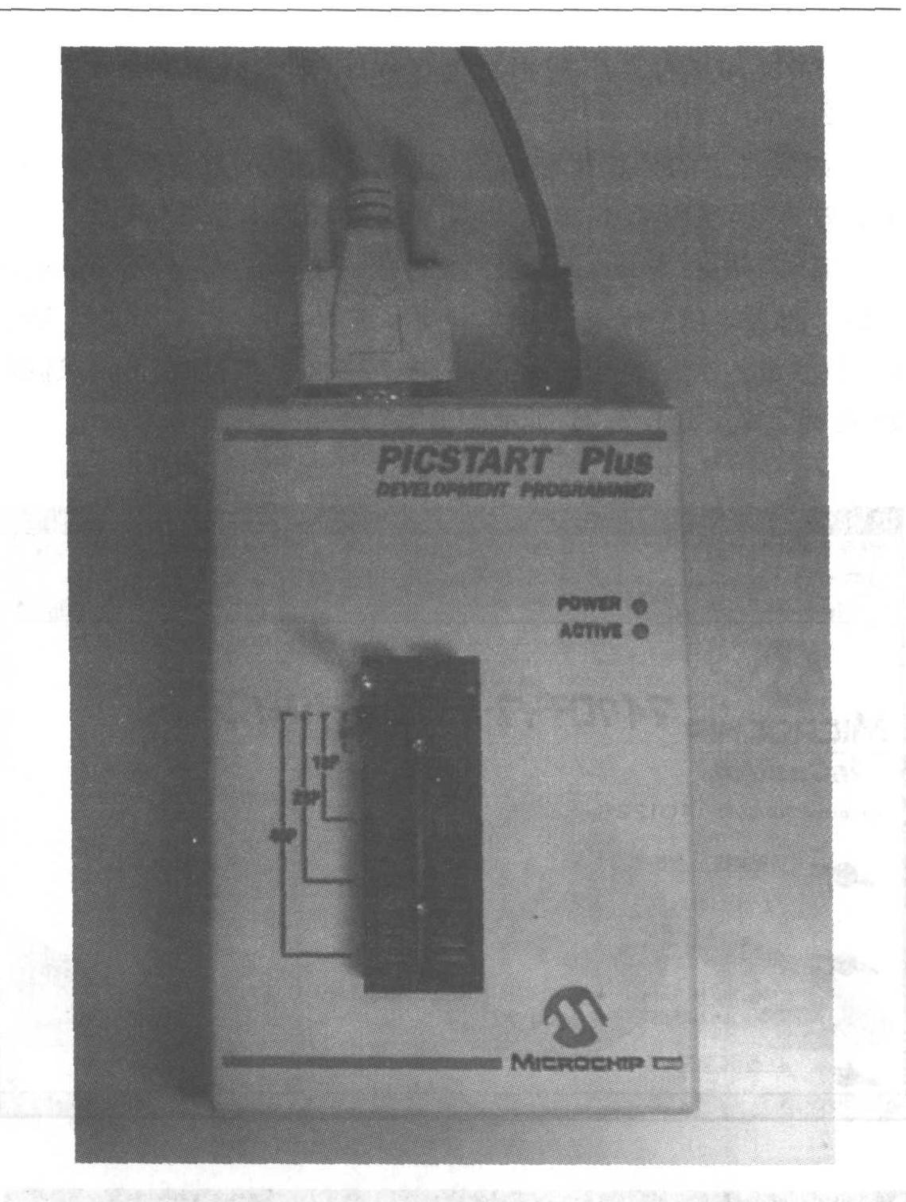


图 2.18 PICSTART Plus

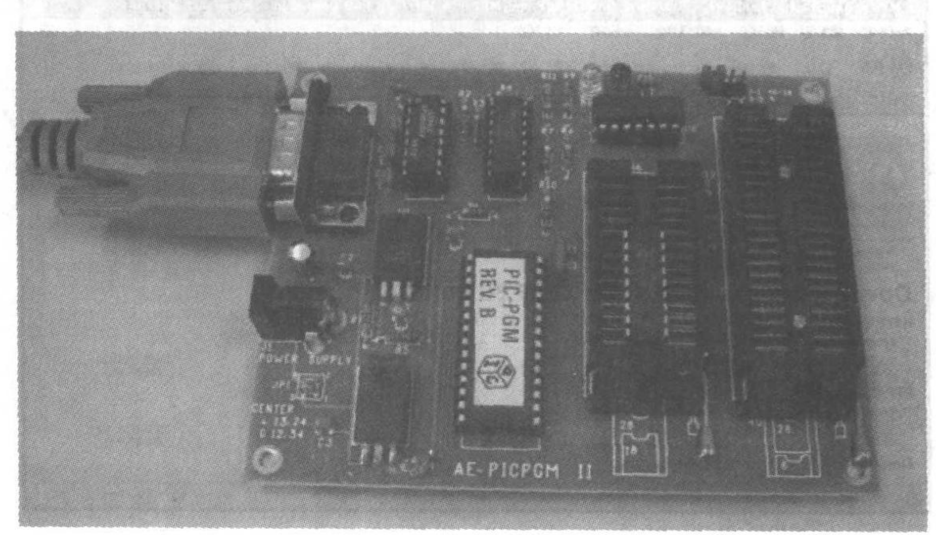


图 2.19 AKI-PIC 编程器

另外，如果打算用 C 语言编写程序，那么必须准备 C 编译器(参见 2.2.5 节)。

2. MPLAB

如果身边的计算机装有合适的 Windows 系统版本，又有了 PIC，则可通过以下方法获得 MPLAB(2003 年 8 月的最新版本是 6.30)。

首先，进入 Microchip 公司的主页(http://www.microchip.co.jp)，参见图 2.20。然后，从[开发工具]开始，选择[下载 MPLAB V6.XX 最新版本(ver 6.30)]，进入美国公司总部的主页。在这个页面的下面，选择 MPLAB(r)IDEv6.30 全部下载，如图 2.21 所示。

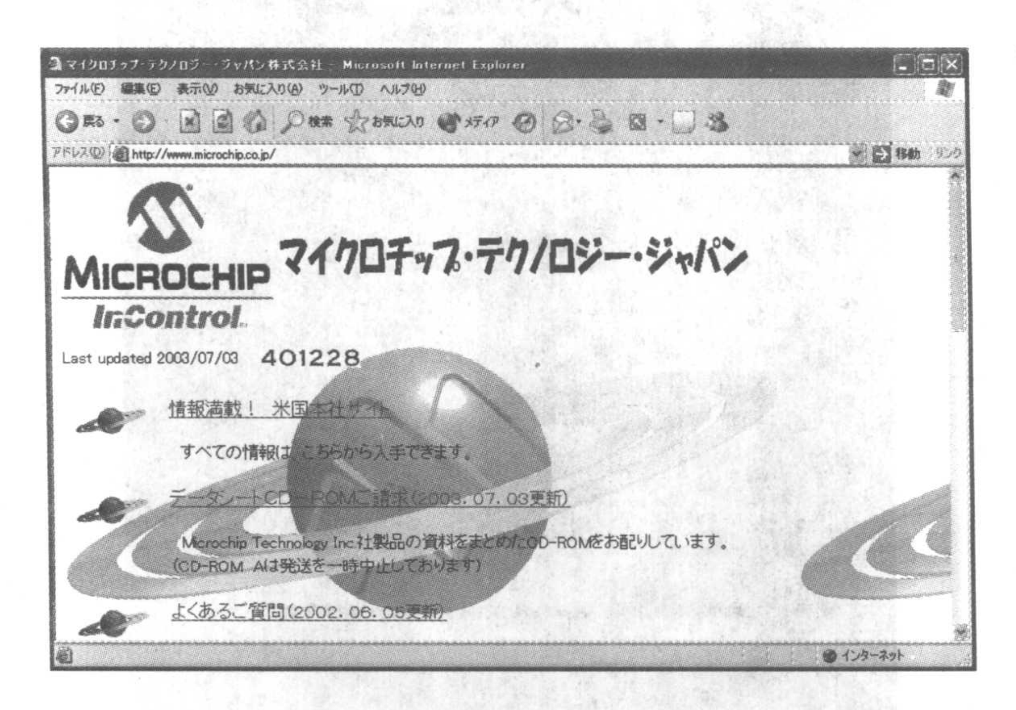


图 2.20　Microchip 公司主页(日本)

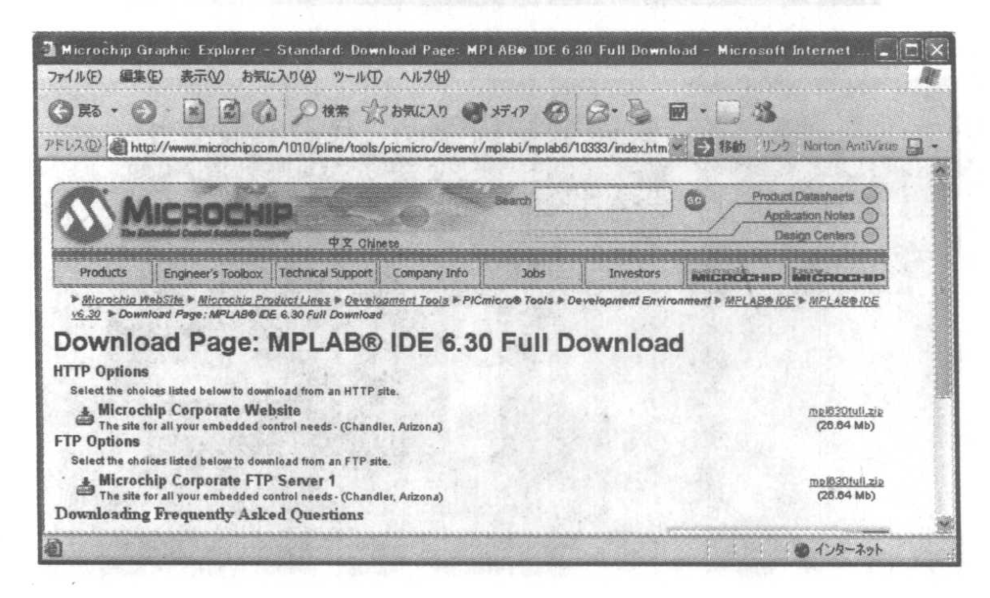


图 2.21　MPLAB 下载页(http://www.microchip.com/1010/pline/tools/picmicro/devenv/mplabi/mplab6/10333/index.htm)

下载文件的大小约 26MB。下载完毕后，双击 MP1630. zip 目录中的执行文件 MP63000. exe 的图标，开始将它安装(参见图 2. 22)。然后出现的所有信息都是英语，按[Next]继续。

在安装过程中，会出现图 2. 23 所示的注册信息，在确认内容后选择[I Agree]，然后单击[Next]。接着继续安装，当安装结束时，屏幕上将出现注意事项的说明，确认后，关闭窗口。

安装结束后，会出现有关并行接口装置或 USB 驱动器的设置窗口，因为暂时尚不需要，故可以直接将其关闭。MPLAB 安装后，在计算机的桌面上出现图 2. 24 所示的图标。双击该图标，就可以起动 MPLAB 了。

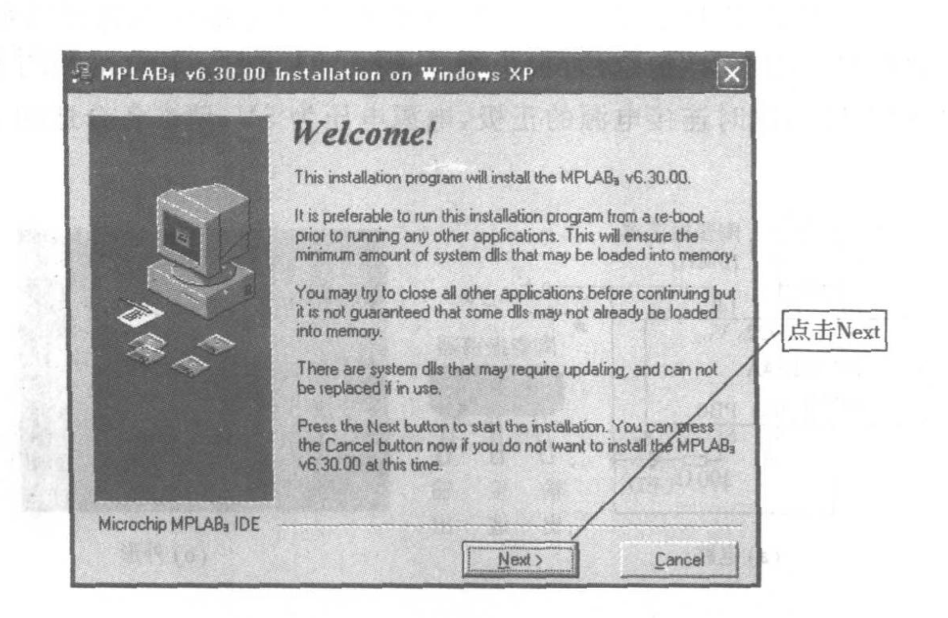


图 2. 22　程序安装开始

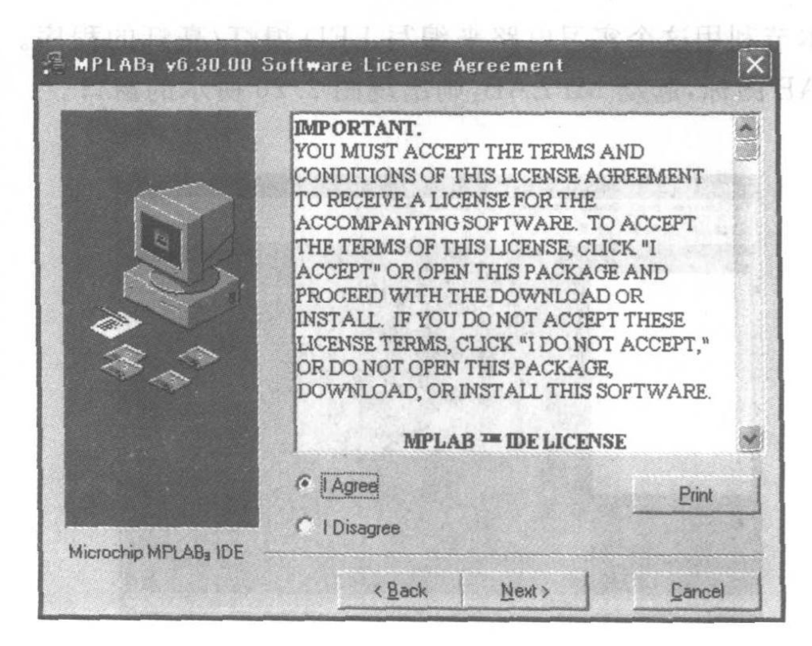


图 2. 23　选择[I Agree]

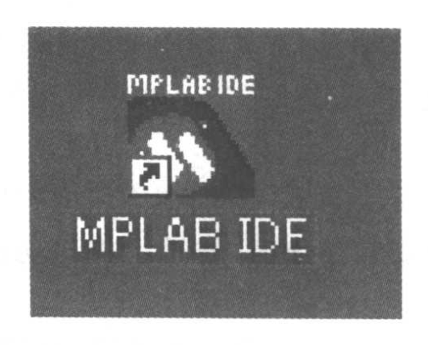
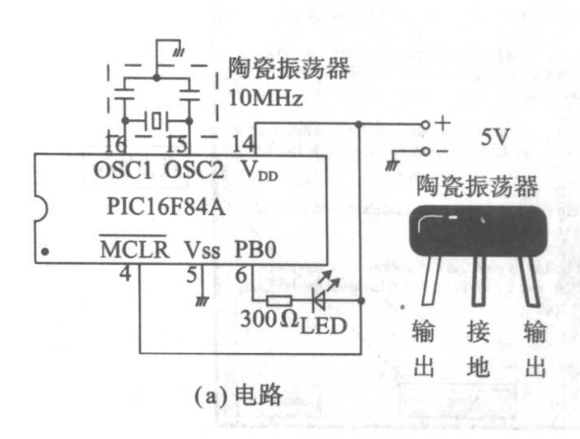


图 2.24　MPLAB 起动图标（安装一结束就显示）

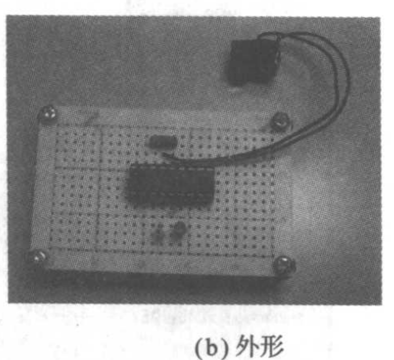
3. 程序开发过程

下面讲解在 MPLAB 软件环境下使用汇编语言进行程序开发的过程。

以图 2.25 所示的 1 个 LED 的简单实习电路为例进行说明。若从引脚 RBO 输出“1”，则 LED 熄灯，若输出为 0，则 LED 亮灯。PIC 的 $\overline{\text{MCLR}}$ 引脚是复位电路用的端口，这时连接电源的正极，电源电压为 5V，请准备合适的 AC 适配器。



(a) 电路



(b) 外形

图 2.25　简单实习电路

本节利用这个实习电路来编写 LED 熄灯/亮灯的程序。双击图 2.24 所示的 MPLAB 图标，起动 MPLAB，则出现图 2.26 所示的窗口。

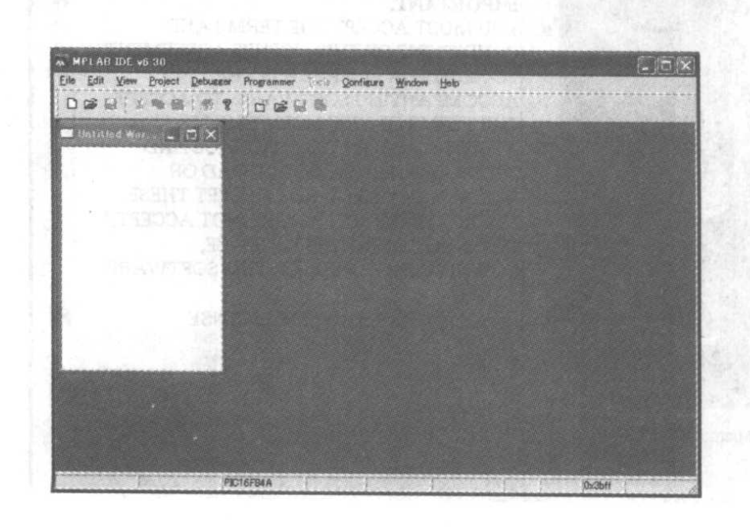


图 2.26　MPLAB 的起动窗口

如图 2.27 所示，选择工具栏的[Project]→[Project wizard]，便出现 Project wizard 起动的窗口。

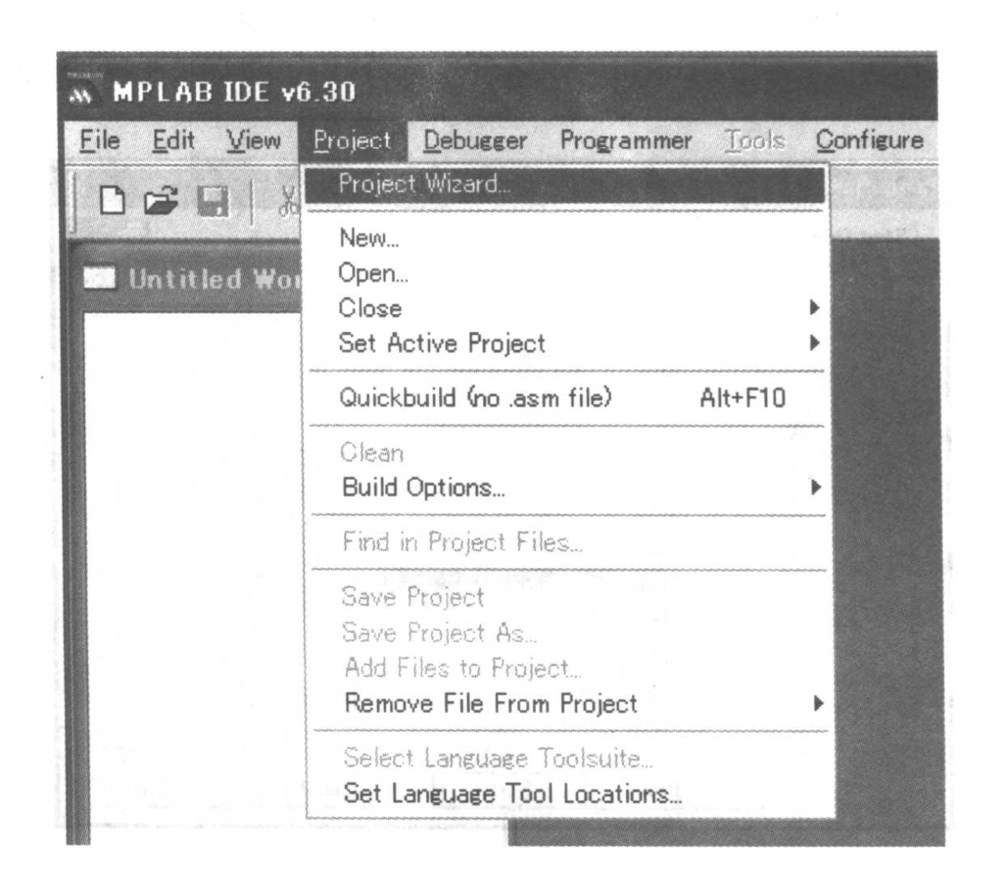


图 2.27 选择[Project]→[Project wizard]

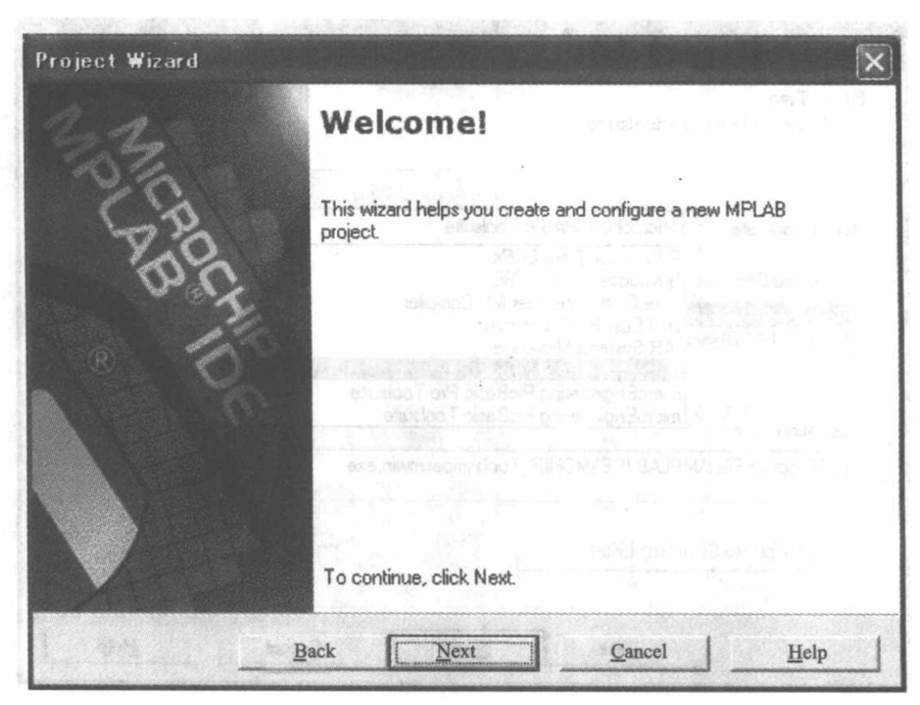


图 2.28 Project wizard

Project wizard 是便于 MPLAB 使用的辅助程序。

在图 2.28 中，单击[Next]按钮，进入下一个窗口。在图 2.29 中，指定 PIC 的类别为[PIC16F84A]，然后，单击[Next]按钮。在图 2.30 的对话框[Active Toolsuite]中，选择[Microchip MPASN Toolsuite]，然后单击[Next]按钮。

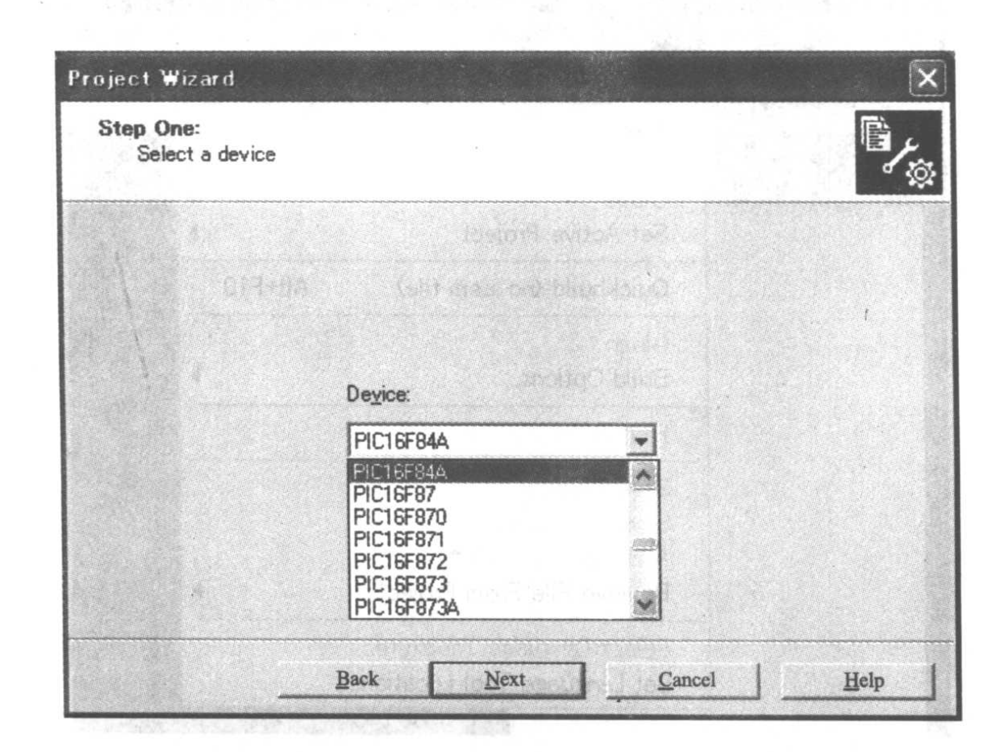


图 2.29　指定[PIC16F84A]

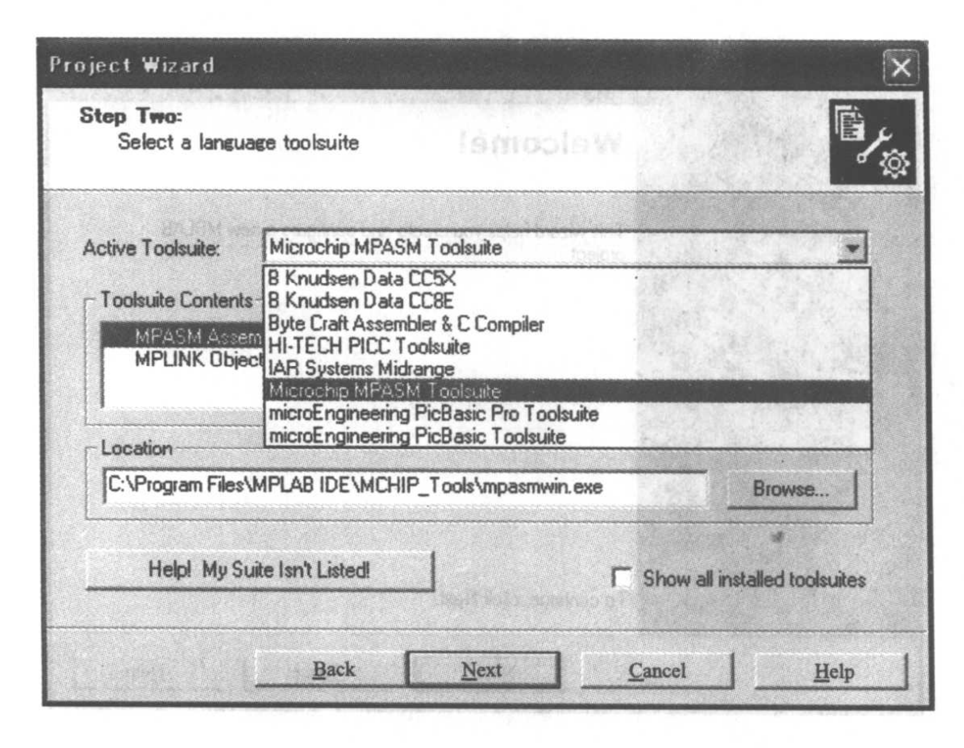


图 2.30　选择[Microchip MPASM Toolsuite]

在 MPLAB 中，由于程序以工程为单位进行管理，所以在图 2.31 中应该指定工程(Project)名和它的目录。这里，已经指定了工程名为[led1]，它对应的目录名为[test]。在作者的计算机上，目录所在的驱动盘是 C 盘，至于读者，应该指定与手头使用的计算机对应的驱动盘的目录。Project 名与目录输入完毕后单击[Next]进入图 2.32。

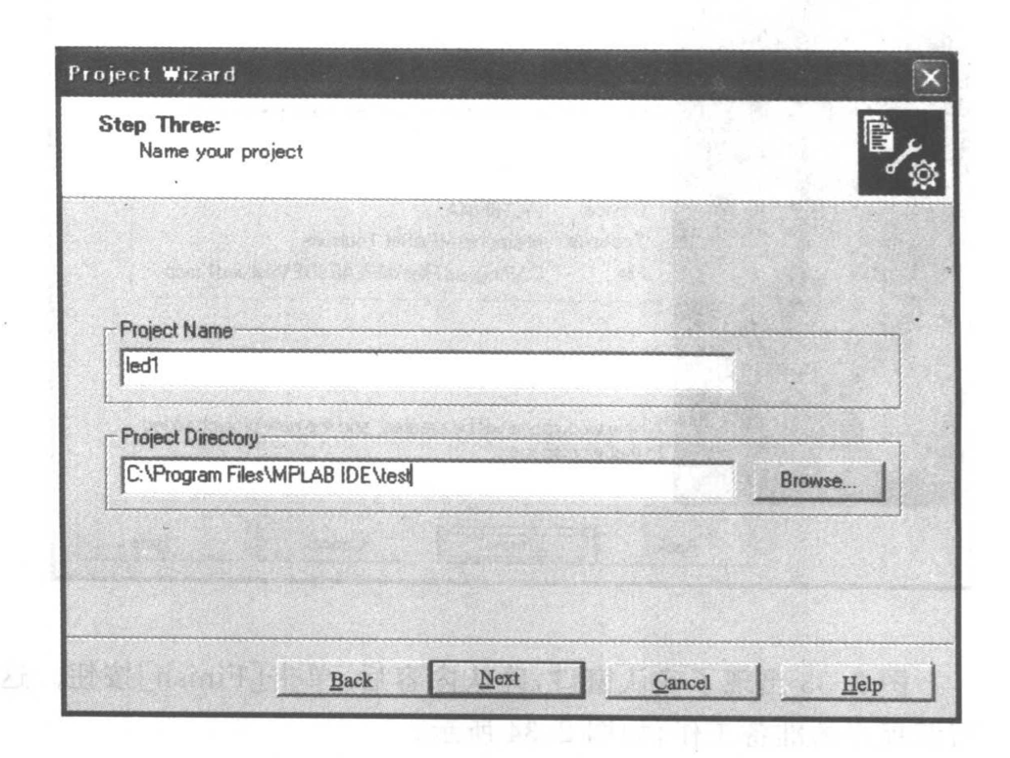


图 2.31 工程名与目录的指定

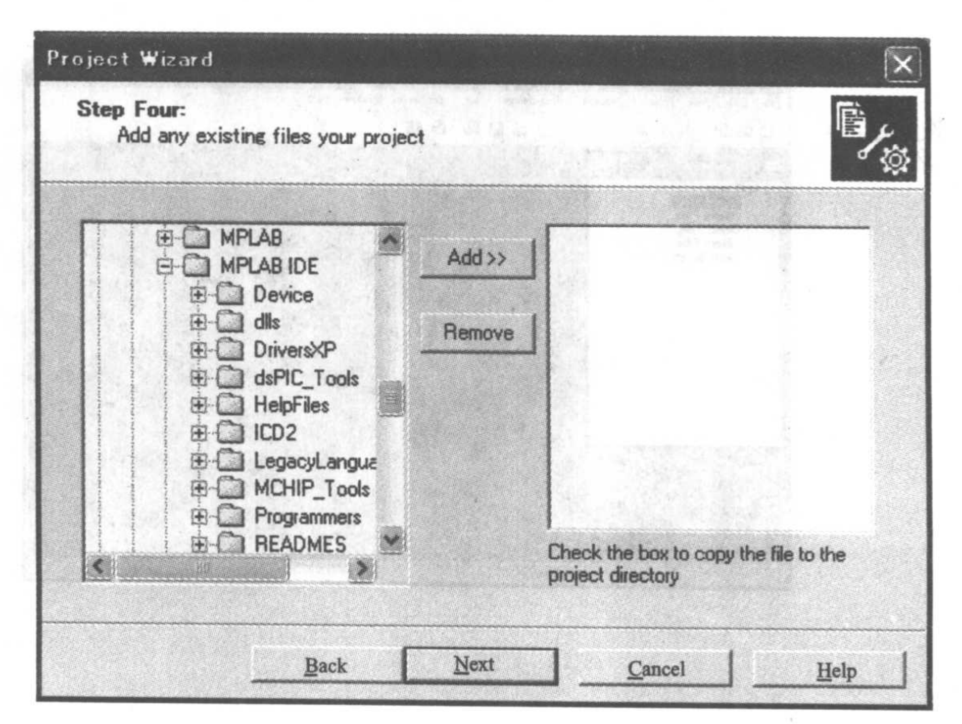


图 2.32 进入[Next]

图 2.32 是注册编写好的源文件的窗口。因为目前源文件尚未编写完毕，故应该随即单击[Next]。

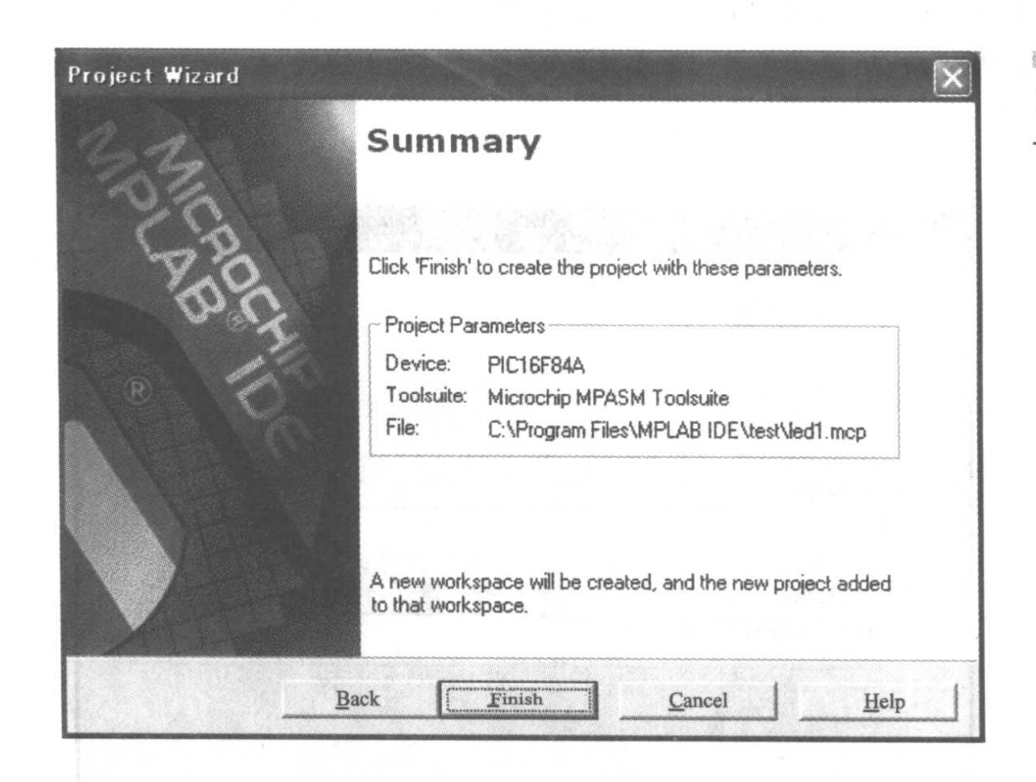


图 2.33　确认窗口

图 2.33 出现了确认窗口，确认内容后，单击[Finish]按钮。这样，就完成了编写源程序的准备工作，如图 2.34 所示。

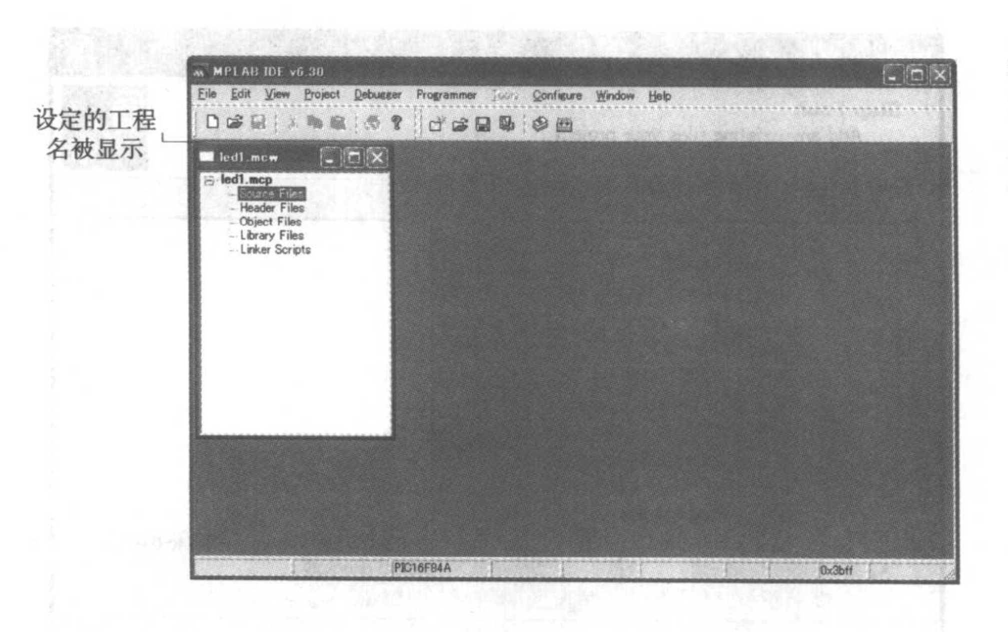


图 2.34　完成编写源程序的准备

```
;* * * * * * * * * * * * * * * * * * * * * * * * *
;程序清单 2.2
;LED 熄灯/亮灯程序
            LIST        P = PIC16F84A     ; 指定使用的 PIC
            INCLUDE     "P16F84A.INC"     ; 指定写入设定文件
;* * * * * * * * * * * * * * * * * * * * * * * * *
LEDD1       EQU     00H                   ; LED 亮灯数据的指定
LEDD2       EQU     01H                   ; LED 熄灯数据的指定
CNT1        EQU     0CH                   ; 定时器 1 的计数变量
CNT2        EQU     0DH                   ; 定时器 2 的计数变量
CNT3        EQU     0EH                   ; 定时器 3 的计数变量
;* * * * * * * * * * * * * * * * * * * * * * * * *
            ORG         0                 ; 存储程序的起始地址

            BSF         STATUS,RP0        ; 选择存储单元的第 1 列
            CLRF        TRISB             ; 设定端口 B 全部为输出模式
            BCF         STATUS,RP0        ; 选择存储单元的第 0 列

REPEAT      MOVLW       LEDD1             ; 把亮灯数据置于 W 寄存器中
            MOVWF       PORIB             ; 向端口 B 输出亮灯数据

            CALL        TIMER3            ; 调用 1 秒定时器

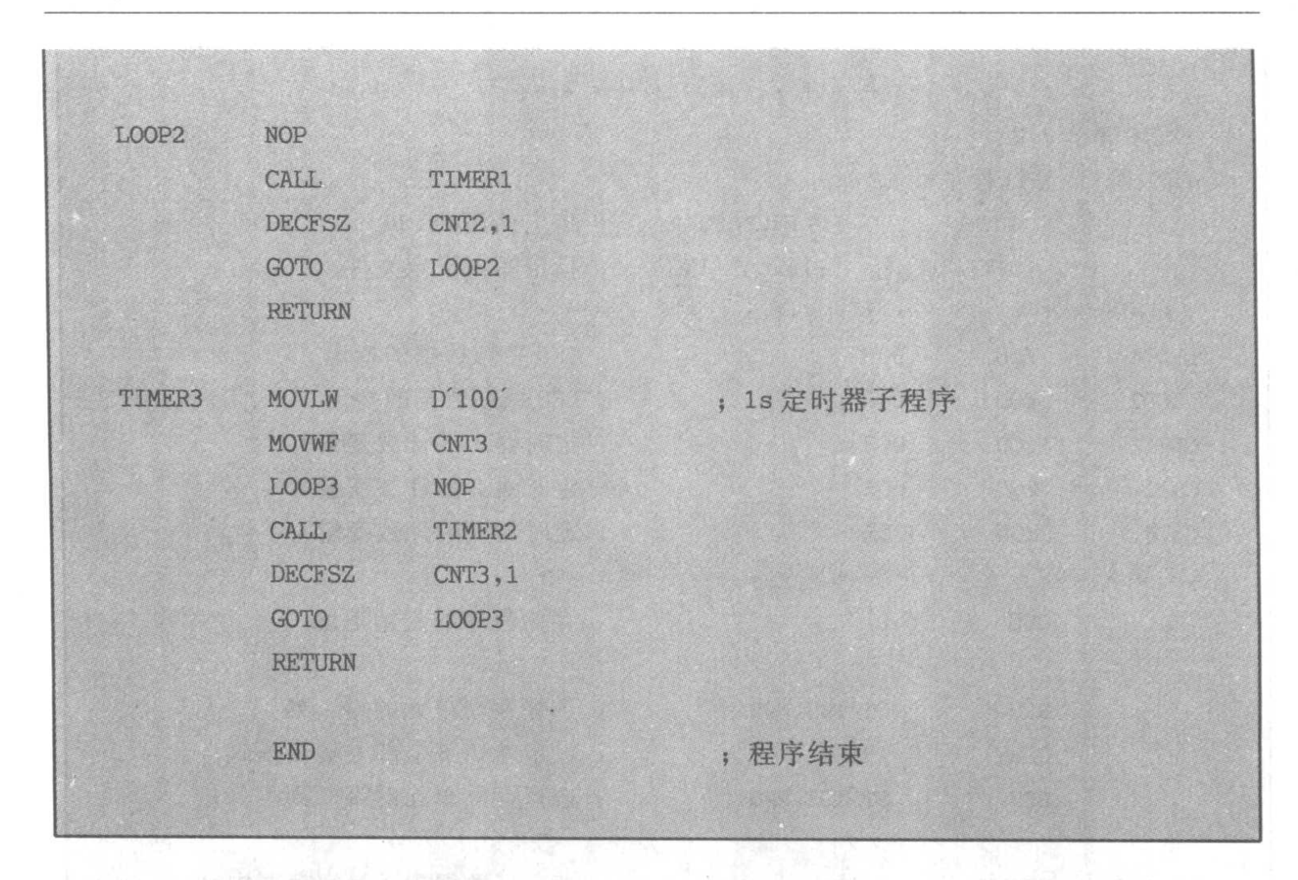
            MOVLW       LEDD2             ; 把熄灯数据置于 W 寄存器中
            MOVWF       PORTB             ; 向端口 B 输出熄灯数据

            CALL        TIMER3            ; 调用 1 秒定时器

            GOTO        REPEAT            ; 重复操作

TIMER1      MOVLW       D'62'             ; 0.1ms 定时器子程序
            MOVWF       CNT1
LOOP1       NOP
            DECFSZ      CNT1,1
            GOTO        LOOP1
            RETURN

TIMER2      MOVLW       D'100'            ; 10ms 定时器子程序
            MOVWF       CNT2
```



```
LOOP2       NOP
            CALL        TIMER1
            DECFSZ      CNT2,1
            GOTO        LOOP2
            RETURN

TIMER3      MOVLW       D'100'              ；1s 定时器子程序
            MOVWF       CNT3
            LOOP3       NOP
            CALL        TIMER2
            DECFSZ      CNT3,1
            GOTO        LOOP3
            RETURN

            END                             ；程序结束
```

程序清单 2.2　LED 亮灯/熄灯程序(汇编语言)

下面来说明编写源程序并向 PIC 传送的操作过程。图 2.25 所示的实习电路的 LED 熄灯/亮灯的程序如程序清单 2.2 所示。

如图 2.35 所示，从工具栏选择[File]→[New]，出现编辑窗口后向里面输入 list2 的源程序，如图 2.36 所示。在编程中，注释语句可以使用日语。

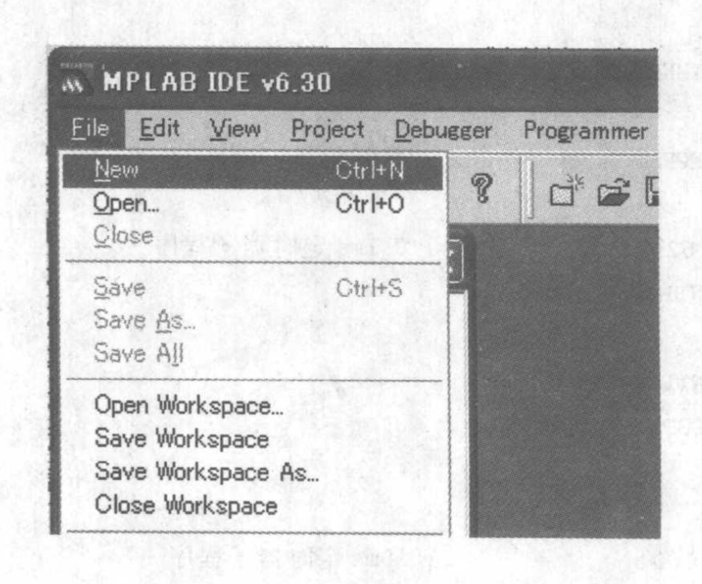


图 2.35　选择[File]→[New]

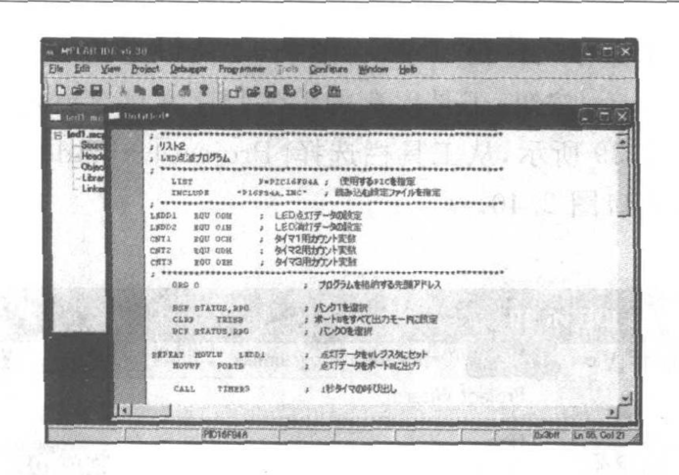


图 2.36 输入源程序

源程序输入完成后，如图 2.37 所示，从工具栏中选择[File]→[Save As]就会出现图 2.38 所示的保存文件用窗口。

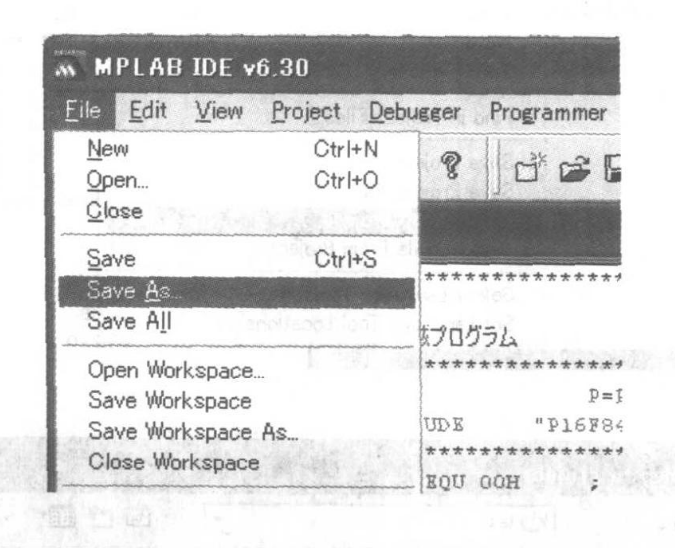


图 2.37 选择[File]→[Save As]

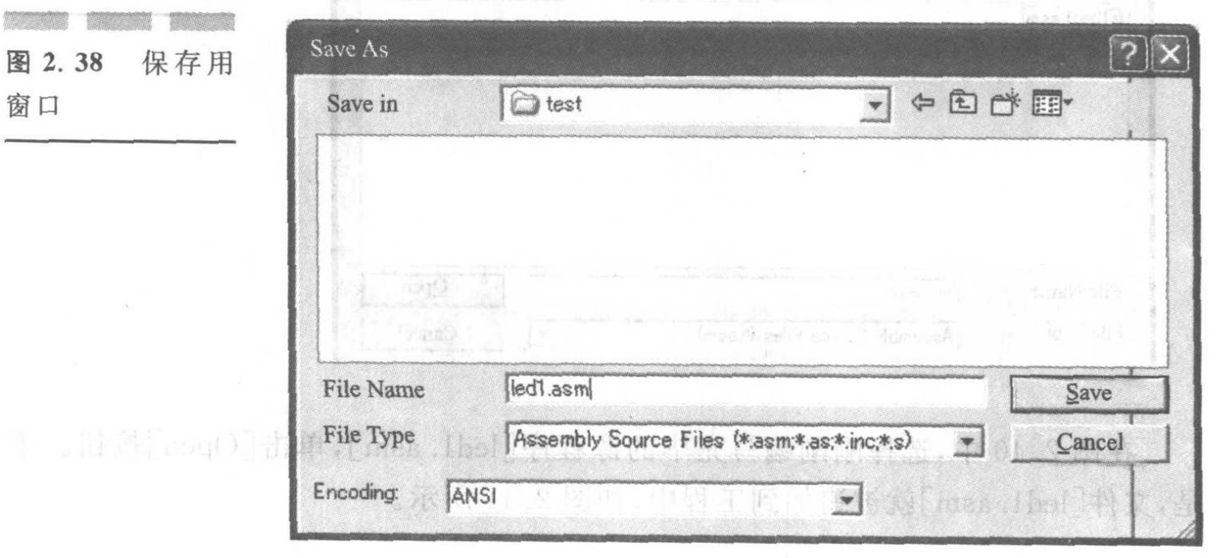


图 2.38 保存用窗口

在这个保存文件窗口中，将图 2.31 所设定的 test 目录和文件名指定为[led1.asm]，按[Save]按钮，于是保存的汇编程序文件名就带有了后缀 asm。

如图 2.39 所示，从工具栏选择[Project]→[Add File to Project]，于是出现文件追加用窗口图 2.40。

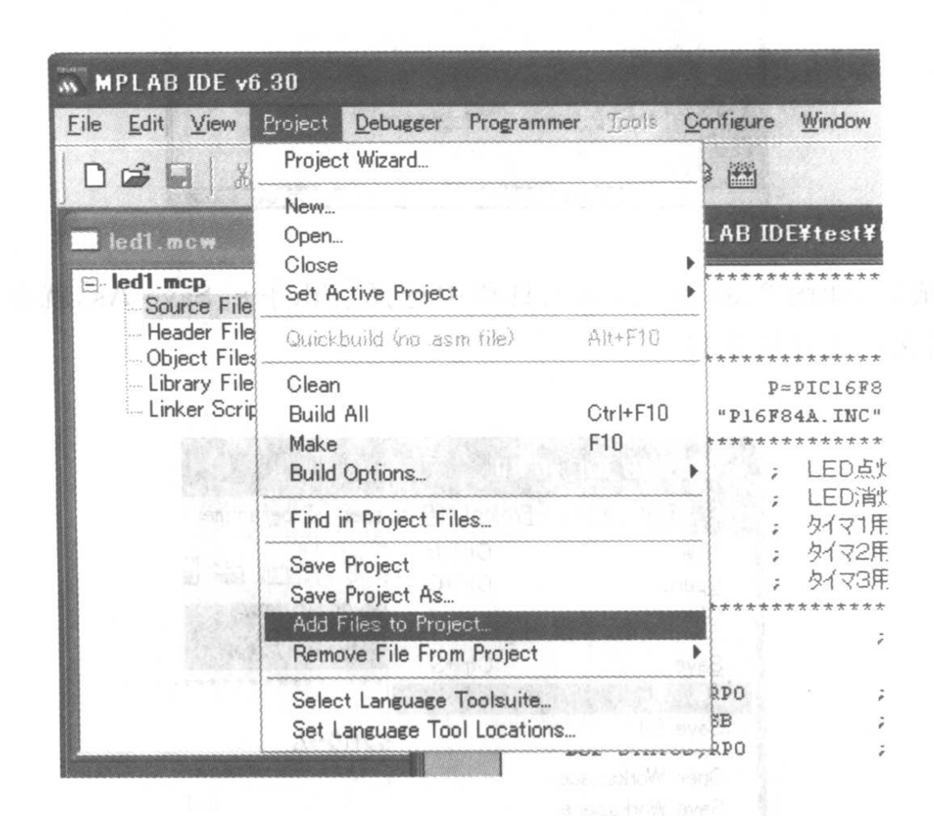


图 2.39　选择[Project]→[Add File to Project]

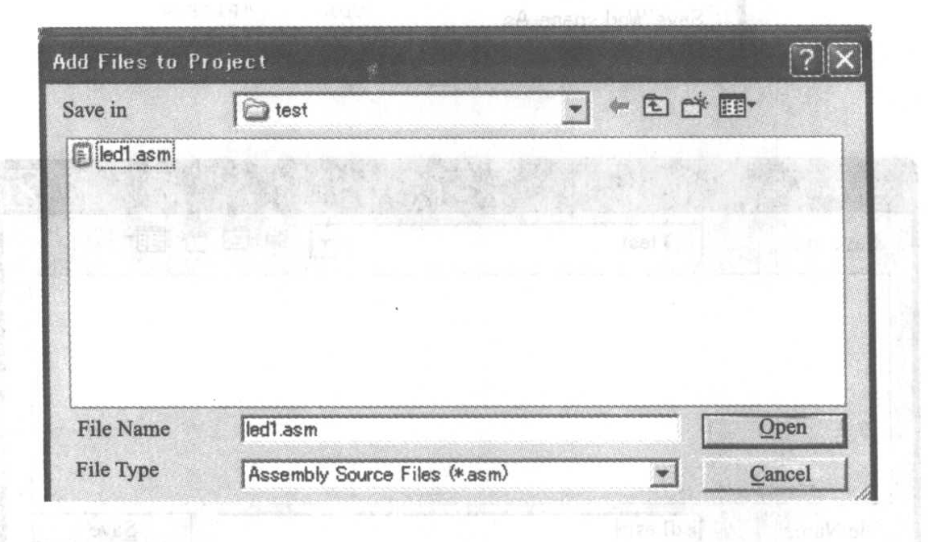


图 2.40　追加文件用的窗口

在图 2.40 中，选择刚刚编写完毕的源程序[led1.asm]，单击[Open]按钮。于是，文件[led1.asm]就被追加到工程中，如图 2.41 所示。

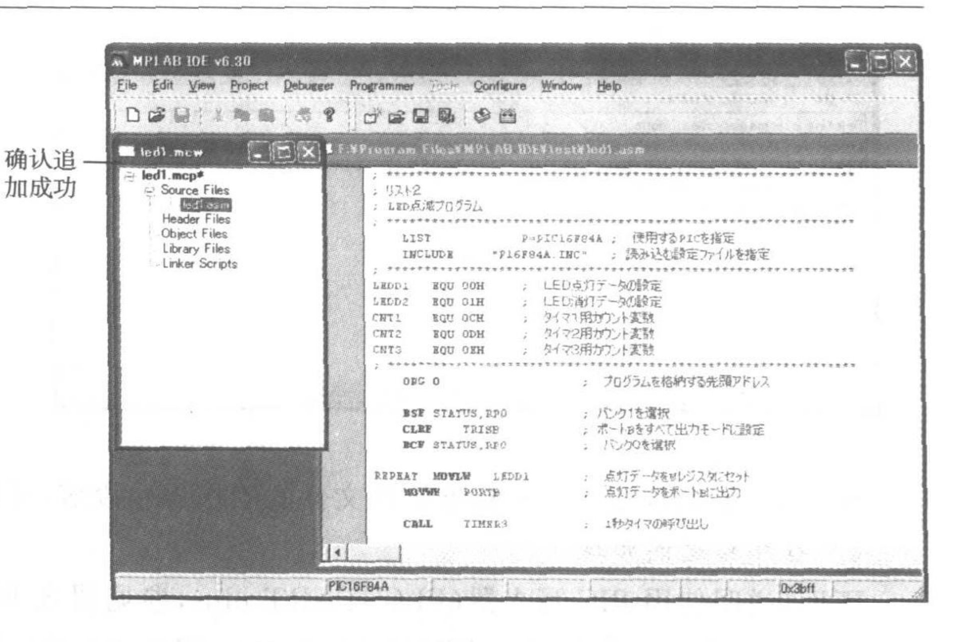


图 2.41 [led1.asm]被追加到工程中

下面,讲解把用汇编语言编写的程序[led1.asm]用汇编编译器生成机器语言。

如图 2.42 所示,从工具栏中选择[Project]→[Build All],于是编译过程开始。

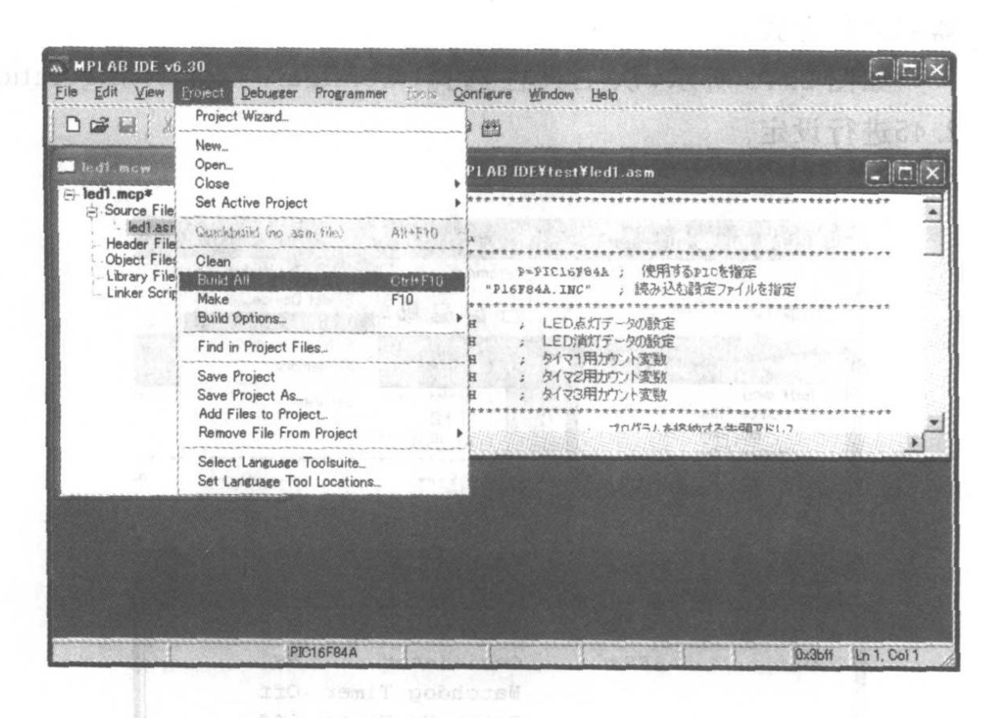


图 2.42 选择[Project]→[Build All]

如果源程序没有错误,就应该出现图 2.43 所示的输出窗口[BUILD SUCCEEDED]。如果有 Error 或 Warring 的显示出现,那么应该返回编辑器窗口(参见图 2.36),修改错误。双击错误出现的地方,会跳到程序中与错误有关的部分。

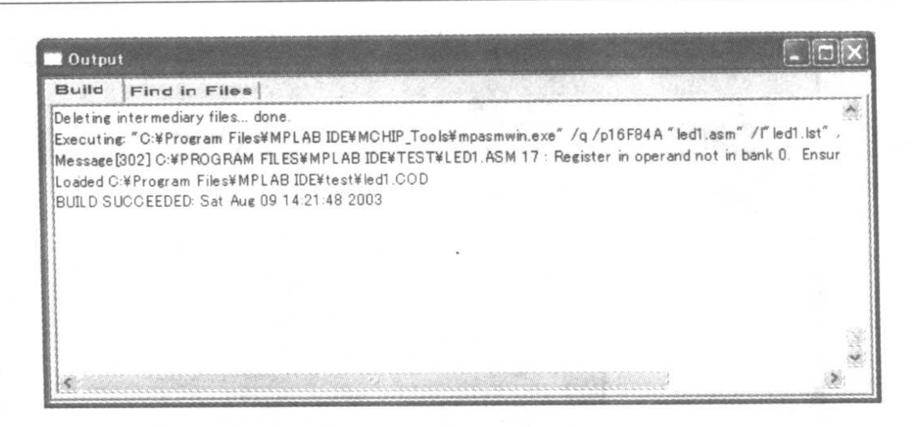


图 2.43　编译完成

至此，用汇编编译器生成机器语言文件的工作告一段落，可以向 PIC 传送了。

4. 程序传送的步骤

下面，说明使用 PIC 写入器(PLCSTART plus，参见图 2.18)实现程序传送的步骤。此前的准备工作是通过 RS232C 串行口把 PICSTART plus 与计算机连接，并将 PICSTART Plus 接入电源。

首先，设定配位，也就是设定使用的时钟振荡器模式以及接通电源后上电定时器的工作方式。

如图 2.44 所示，从工具栏选择[Configure]→[Configuration Bits]，按照图 2.45进行设定。

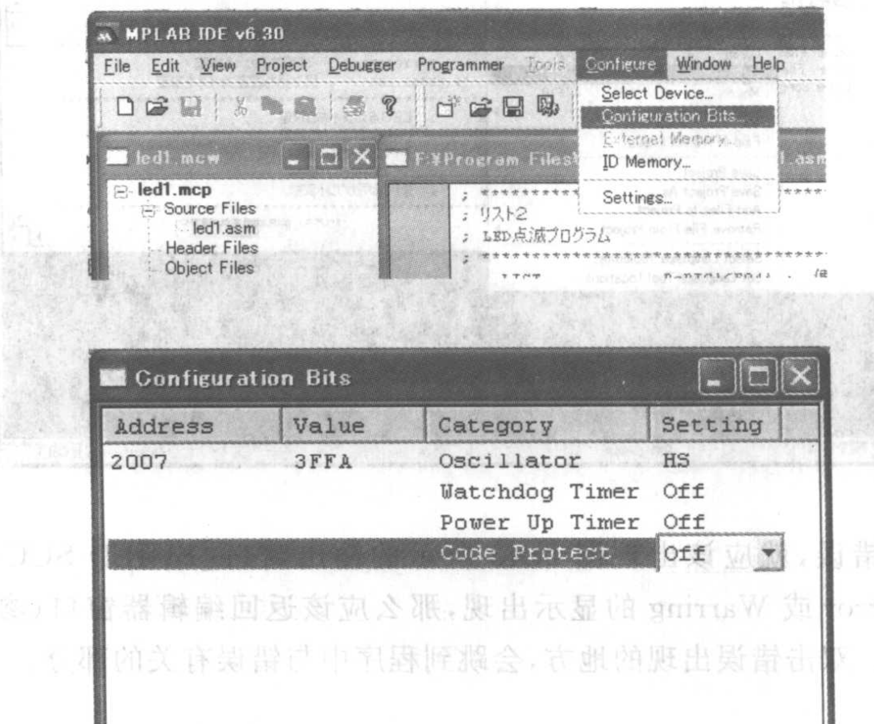


图 2.44　选择[Configure]→[Configuration Bits]

图 2.45　设定配位

设定完成后，单击窗口左上角的[关闭]按钮，关闭窗口。

如图 2.46 所示，从工具栏选择[Programmer]→[Select Programmer]→[PIC-START Plus]。

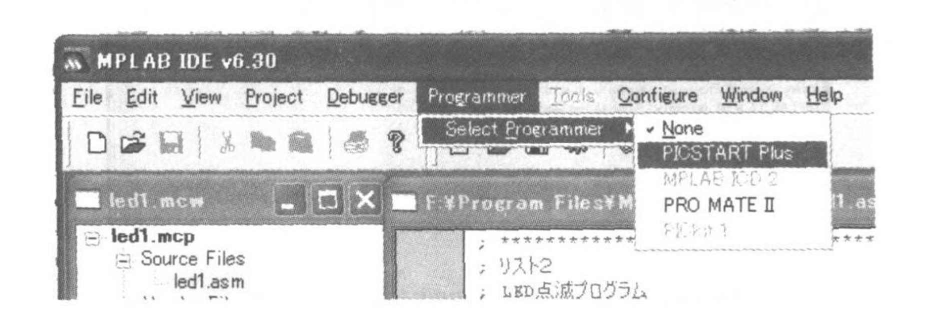


图 2.46 选择[Programmer]→[Select Programmer]→[PIC-START-Plus]

接着，如图 2.47 所示，从工具栏选择[Programmer]→[Enable Programmer]。

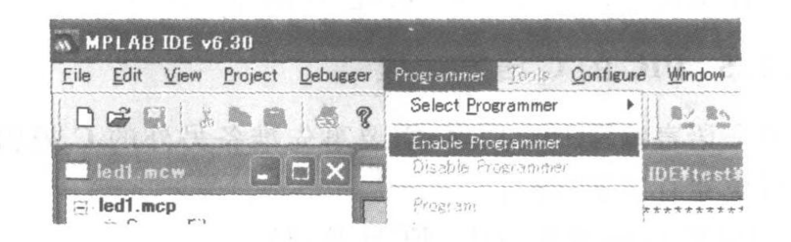


图 2.47 选择[Programmer]→[Enable Programmer]

如图 2.48 所示，从工具栏选择[Programmer]→[Program]，然后就可以开始向 PIC 传送程序。

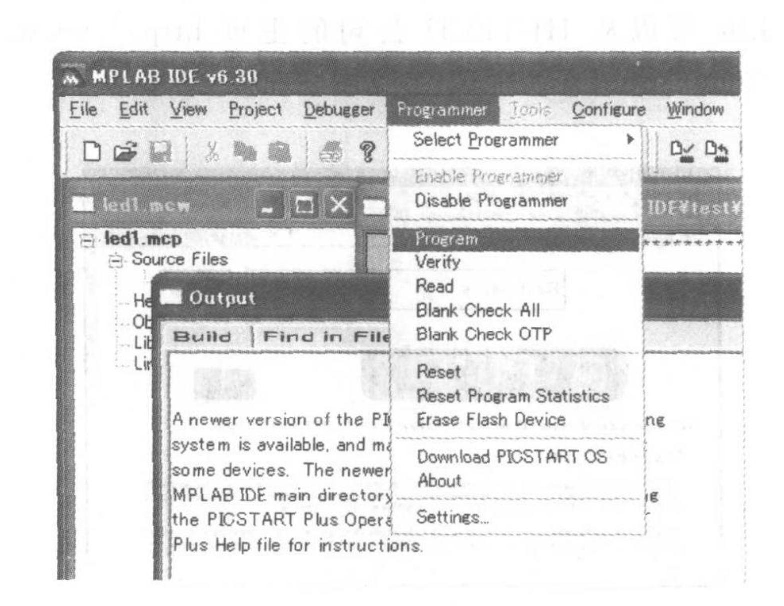


图 2.48 选择[Programmer]→[Program]

如果不发生意外，当程序传送结束后，图 2.49 所示的输出窗口应该显示出[Programming/Verification Completed Successfully!]的字样。

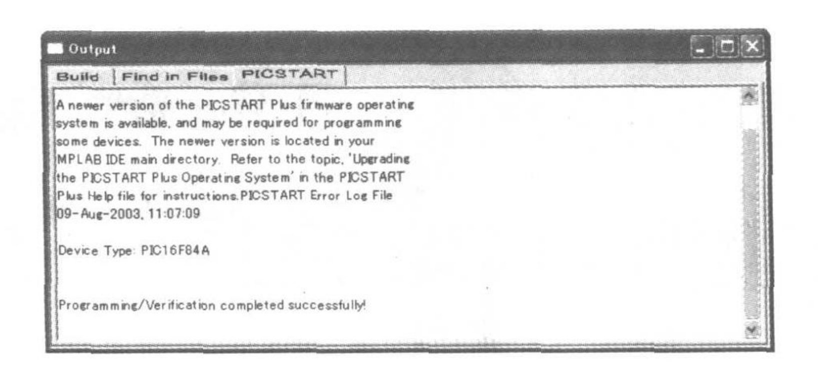


图 2.49　程序传递结束

下面所应该做的事情是把 PIC 安装到实验板的插座中，接通 5V 电源，LED 就会大约以 1 秒钟的间隔重复亮灯/熄灯。

工程完成之后，从工具栏选择[Project]→[Close]。

2.2.5　PIC 的 C 语言编程

用 C 语言编写 PIC 程序，需要事先准备另外的 **C 编辑器**。目前，流行的 PIC 的 C 编辑器有以下几种：

- PICC Lite(免费，HI-TECH 公司)
- PIC C Grich 简装版(2000 日元，秋月电子通商)
- CCS C(24 900 日元，Data Dynamics 公司)

由于 CCS C 编译器具有函数多、性能好等优点，故可以用于实际程序的开发。不过本节以免费的、使用最为广泛的 PICC Lite 为例来说明程序开发的步骤。**PICC Lite** 可以从 HI-TECH 公司的主页 http://www.htsoft.com/products/piclite/index.html 免费下载(参见图 2.50)。

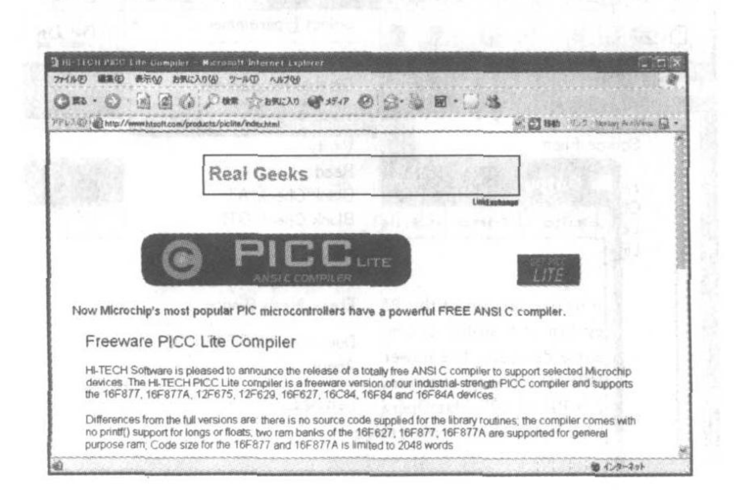


图 2.50　HI-TECH 公司主页

在作者撰写本书时，PICC Lite的最新版本为v8.02。由于旧版本的PICC Lite可以安装到Windows XP计算机，而新版本无法安装到Windows XP计算机，因此，为了使用最新版本，必须使用Windows 98计算机系统。本节先把最新版本安装到Windows 98计算机生成PICCLITE目录，然后将该目录复制到Windows XP计算机，以便使用PICC Lite v8.02。

无论如何，一定要把名为［PICCLIFE］的文件安装到本地磁盘的目录（本书为C盘）中。PICC Lite可以与MPLAB一起进行编程。

下面来尝试将程序清单2.3列写的LED点灯/熄灯C语言程序传送到PIC。

```
// 程序清单 2.3
// LED 点灯/熄灯程序(C 语言)

#include<pic.h>                          // 头文件

main()
{
        long c;
        TRISB = 0x00;                    // 设置端口 B 为输出
        while(1){

                PORTB = 0x01;            // LED 熄灯
                for(c = 0;c<60000;c + +){   // 延时
                        ;
                }
                PORTB = 0x00;            // LED 点灯
                for(c = 0;c<60000;c + +)}   // 延时
                        ;
                }
        }
}
```

程序清单2.3 LED点灯/熄灯程序(C语言)

由上可知，在基本的操作过程中，仅仅是使用语言工具的定义不同，其他与汇编语言相同。起动MPLAB，打开Project Wizard，设置PIC为PIC16F84A（参见图2.27～图2.29）。

在图2.30中，设置语言工具为MPASM（汇编语言），而在图2.51中，语言工具应

设置为[HI-TECH PICC Toolsuite]。然后选中[Toolsuite Contents]窗口中的第 3 项[Location]，指定所有[PICCLITE]目录的文件夹为[picl. exe](参见图 2.52)。

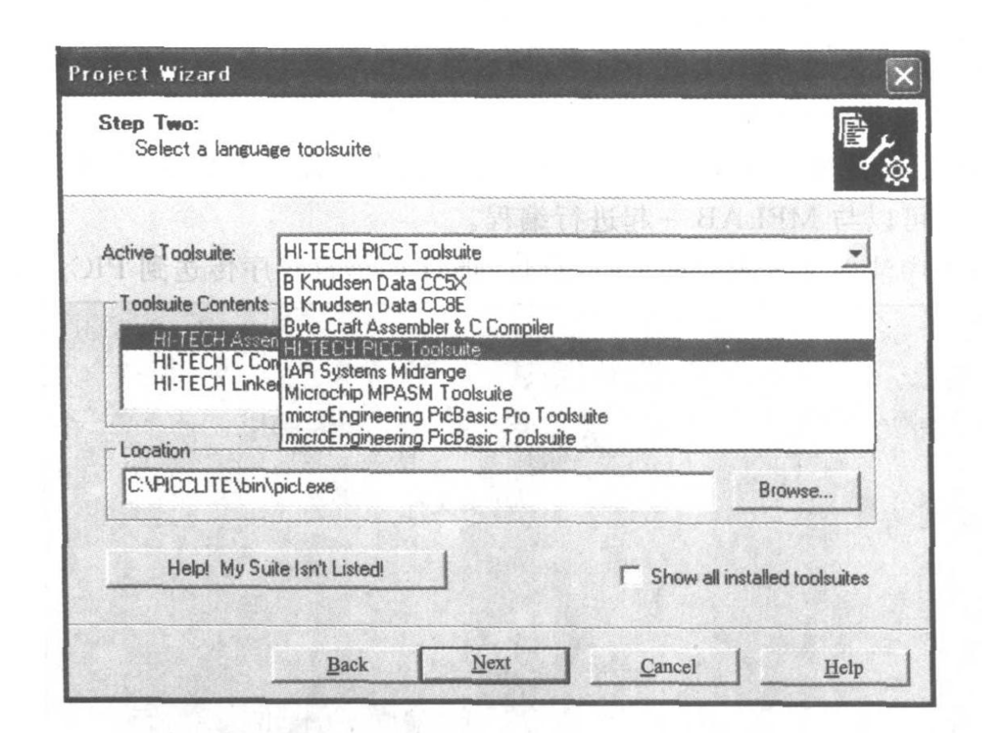


图 2.51　HI-TECH PICC Toolsuite

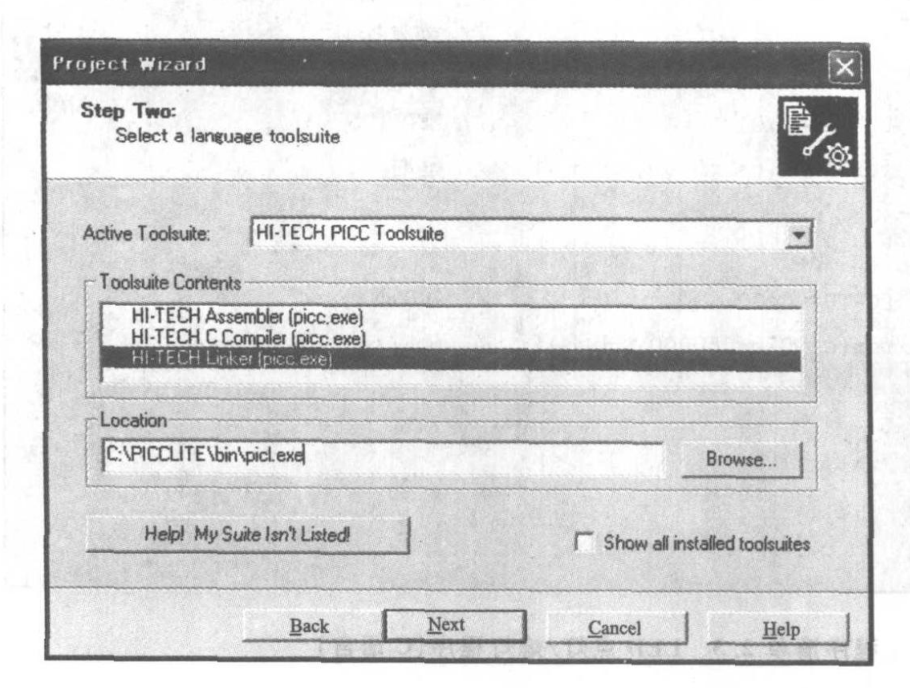


图 2.52　设置[picl. exe]

完成这个设置的另一种方法是在 MPLAB 起动后，从工具栏选择[Project]→[Select Language Toolsuite]，如图 2.53 所示。

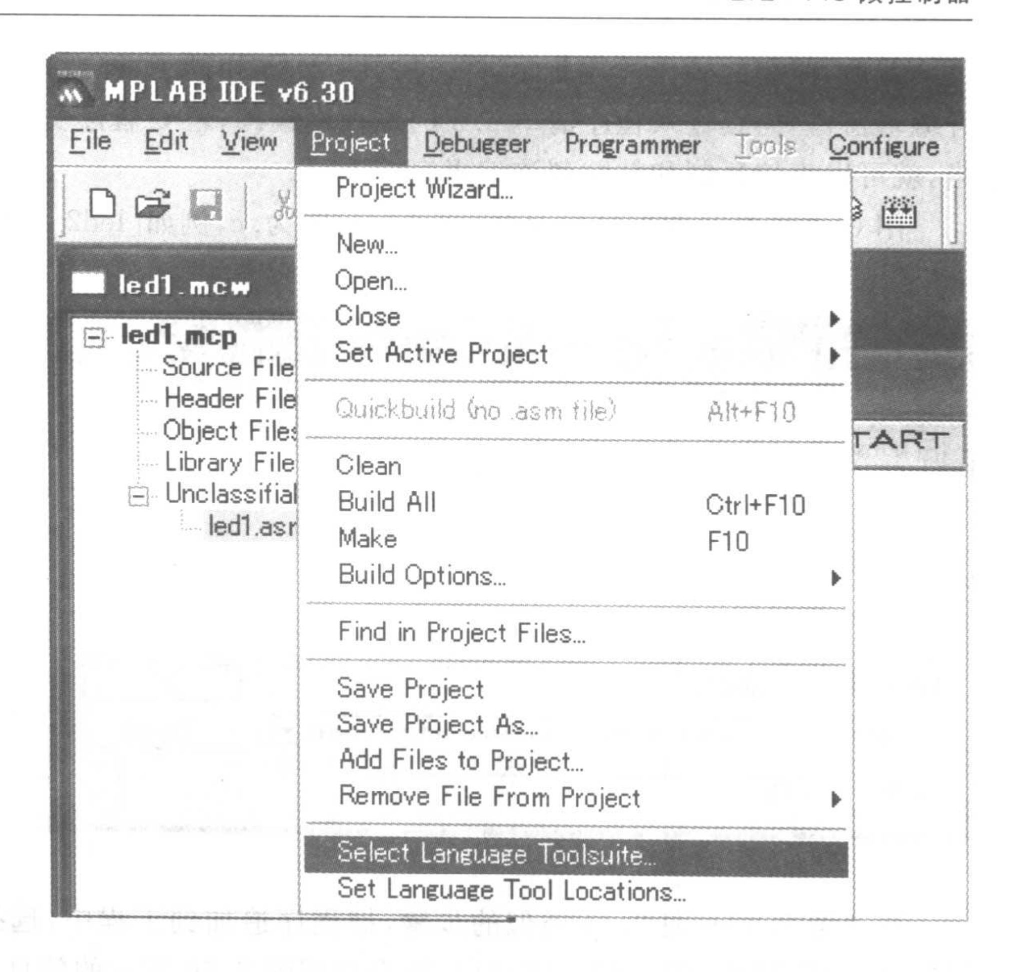


图 2.53　另一种设置方法

图 2.54 列举了工程名与目录名的设置方法。

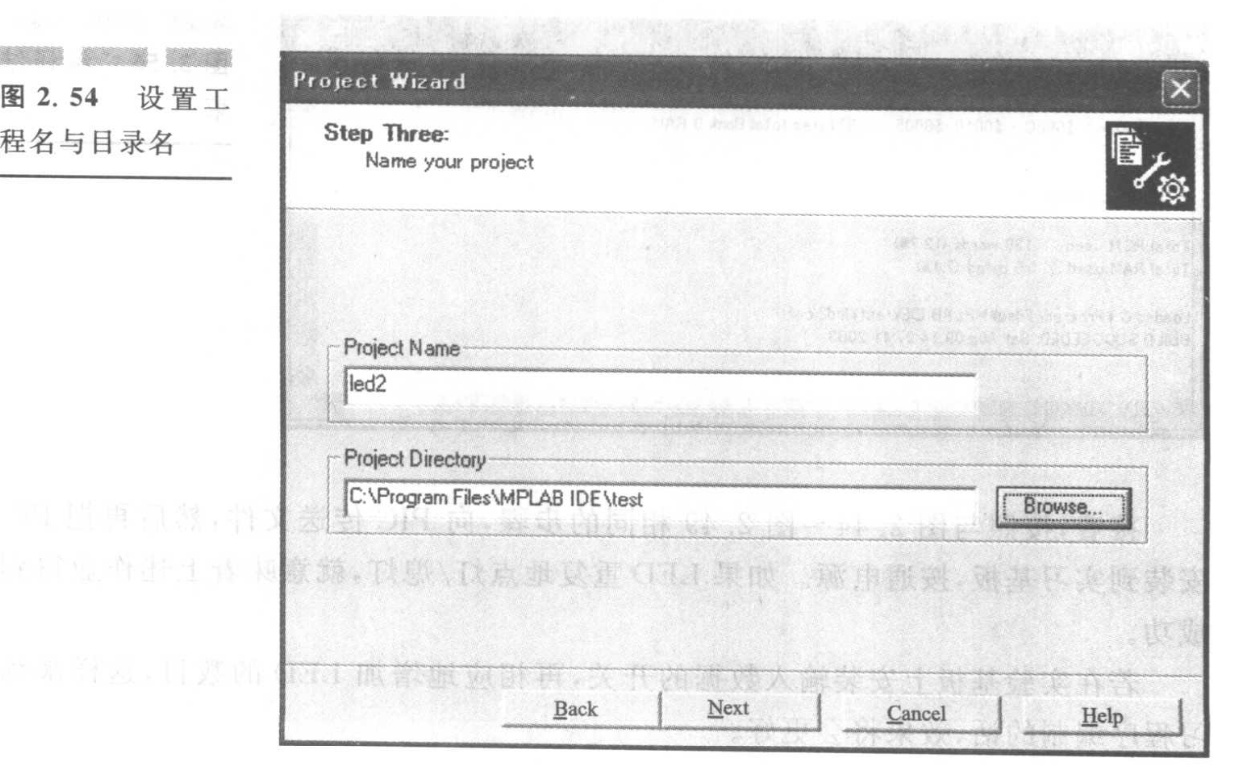


图 2.54　设置工程名与目录名

此后，所有的操作都与前面讲解过的图 2.35 相同了，即选择[File]→[New]打开编辑窗口，向其输入程序清单 2.3 的源程序编码，若 **C 程序**文件原先已经准备妥当，就可以直接复制和粘贴到该编辑窗口下面。

用 C 语言书写的程序保存时，文件的后缀为.c，例如[led2.c]（参见图 2.55）。

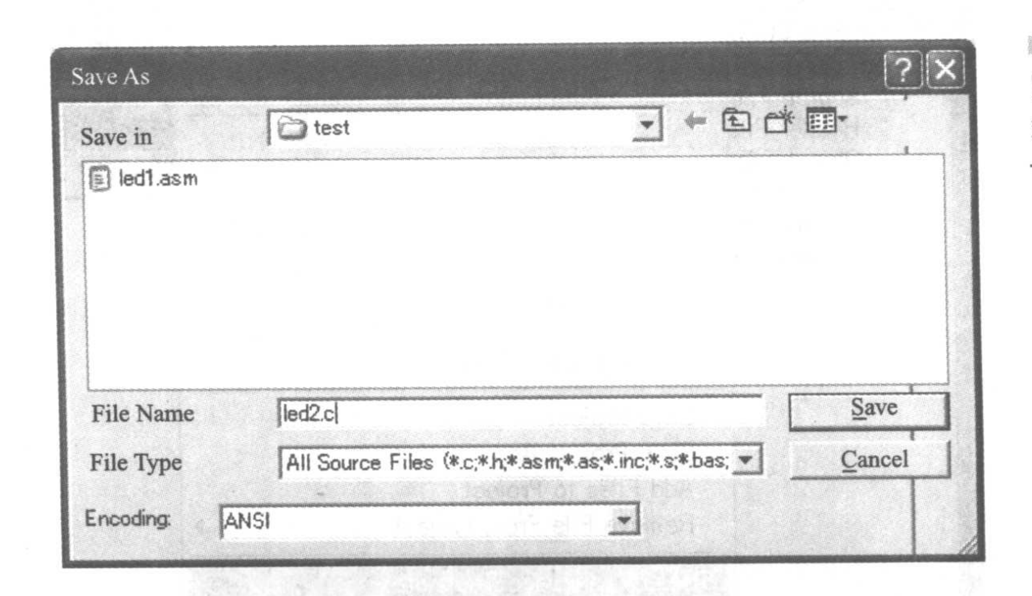


图 2.55　附加 C 程序的后缀[.c]

按照图 2.39～图 2.41 类似的步骤，把程序追加到工程中，选择[Build All]进行编译（参见图 2.42）。编译成功后，就会出现图 2.56 所示的信息。

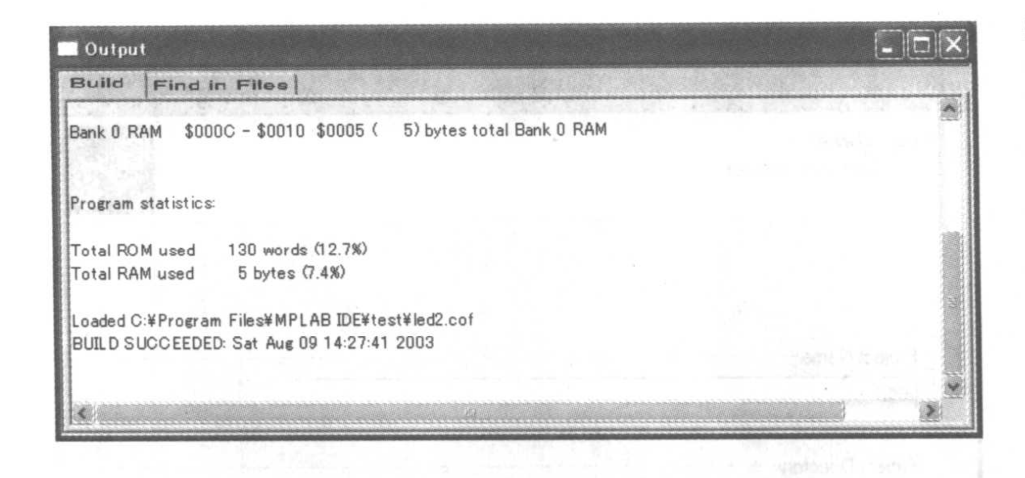


图 2.56　编译结束

接着，按照与图 2.44～图 2.49 相同的步骤，向 PIC 传送文件，然后再把 PIC 安装到实习基板，按通电源。如果 LED 重复地点灯/熄灯，就意味着上述作业得到成功。

若在实验基板上安装输入数据的开关，再相应地增加 LED 的数目，这样来练习程序编制的话，效果将会更好。

2.3 H8 微控制器

H8 是 Renesas Technology 公司生产的一种高性能微控制器，在本节将简要地介绍一下 H8 的构造、特点及其使用方法。

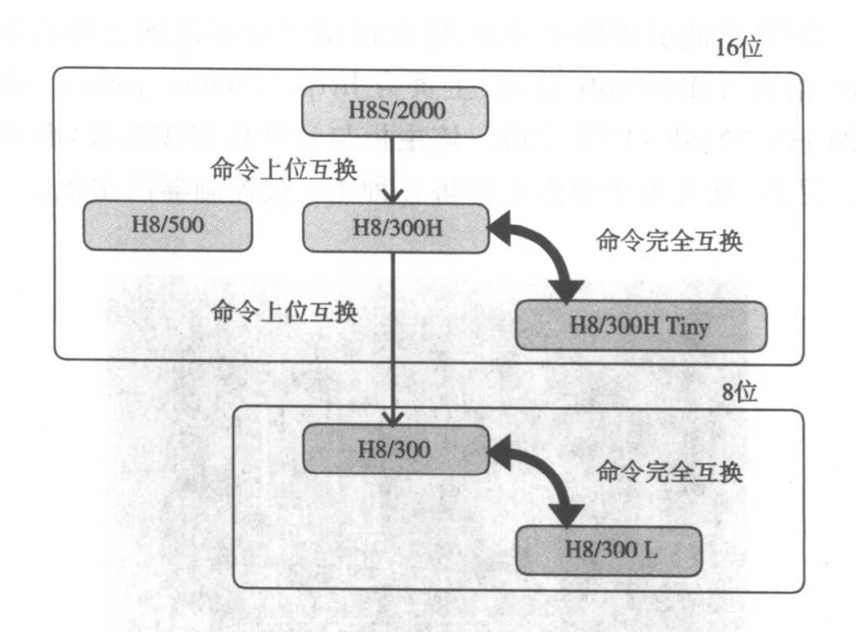
2.3.1 何谓 H8

H8 是一种配备了丰富的周边设备，如各种接口、A/D、D/A 转换器、定时器等单片机。与 PIC 相似，它与 RAM，以及电可擦可写的闪存只读存储器组成的微控制器系统已经广泛地应用于各个领域。如前所述，PIC 属于哈佛型 RISE 方式的微控制器，程序命令的数量较少。对比之下，H8 则属于诺伊曼型的 CISC 方式的微控制器，备有丰富的命令系统。因此一般来说，与 PIC 相比，H8 可用于更高级的控制。

H8 的产品有许多系列。图 2.57 中，H8 分为 6 个系列，每个系列中有多个不同的型号。

在这些系列中，智能玩具最流行的是 H8/300H 和 H8/300H Tiny 系列。由于这两种系列使用的是同一个 H8/300 CPU，所以两者指令可以完全可换。H8/300 比 Tiny 系列的周边功能更强，可根据用途不同进行选择。在 H8/300 中有代表性的型号是 H8/3048F、H8/30052F，在 H8/300H Tiny 中是 H8/3664BP。图 2.59 给出了这些型号的外形照片。

图 2.57 H8 的分类



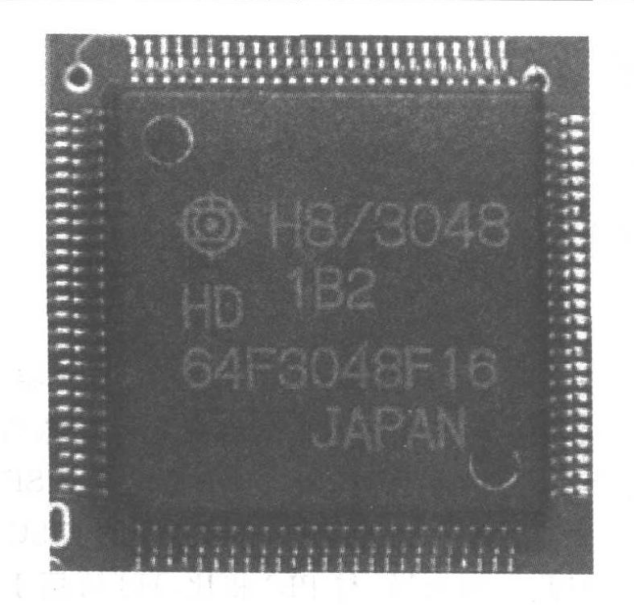


图 2.58　H8/3048F(QFP 型)

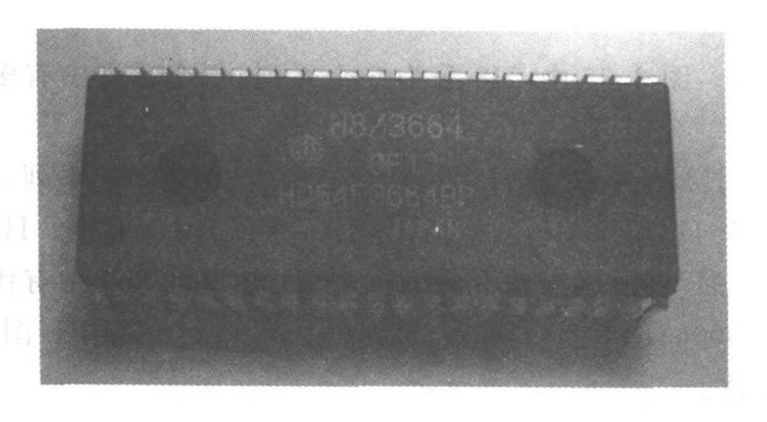


图 2.59　H8/3664BP(SPIP 型)

QFP 型的引脚数比较多，因此使用时最好定购已固定好的 H8 主板成品。图 2.60 给出 **Yellow-soft 公司**(主页为 http://www.yellowsoft.com/index.html)出售的 H8/3048F-ONE 主板。该主板与计算机直接连接，因此进行程序传送非常方便。另外，秋月电子通商也销售各种 H8 微控制器的主板。

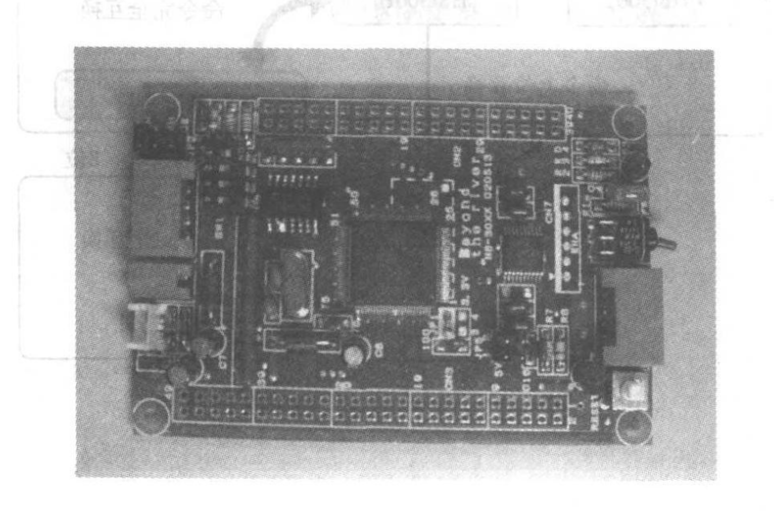


图 2.60　微控制器主板(Yellow-soft 公司提供)

2.3.2 H8的构成

这里，以H8/300H系列的典型型号H8/3048F为例，介绍H8的构成。图2.61表示H8/3048F的主要组成。它的引脚数为100条，超过半数的引脚可以兼做几种功能使用。



图 2.61 H8/3048F的组成（摘自：H8微控制器入门，堀桂太郎著，2003）

H8/3048F的输入/输出端口共有10个（参见图2.62）。它可使用最大78个输入/输出引脚。因此可知，与PIC**16F84A**相比，它的RAM与ROM尺寸增大，并具有各种周边功能。

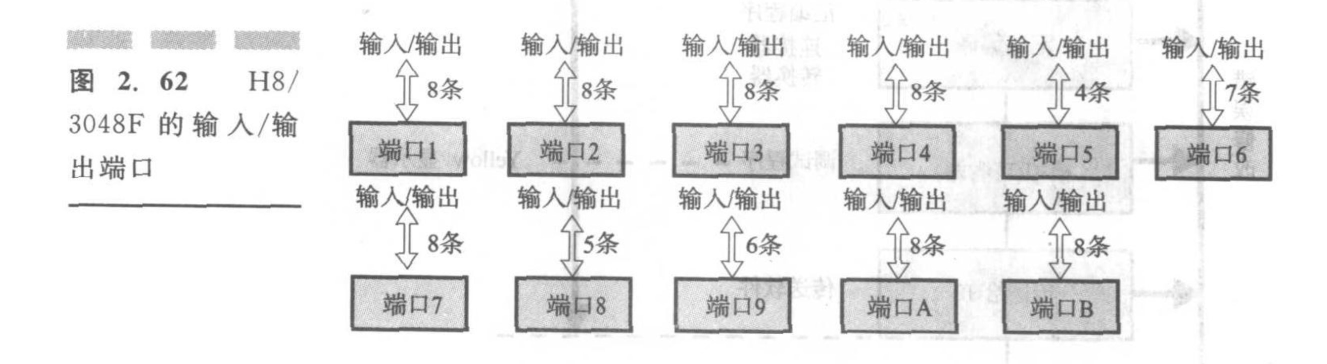


图 2.62 H8/3048F的输入/输出端口

表2.6列出H8/300H CPU中所使用的**汇编命令**。在H8中，即使相同的命令，依照数据长度的差别，使用方法也有不同。表2.6把条件分叉命令统称为BCC。因此，实际的命令数要比表2.6给出的更多一些。

功 能	命 令
数据传送	MOV,PUSH,POP
算术运算	ADD,SUB,ADDX,SUBX,INC,DEC,ADDS,SUBS,DAA,DAS,MULXU,MULXS,DIVXU,DIVXS,CMP,NEG,EXTS,EXTU
逻辑运算	AND,OR,XOR,NOT
移位	SHAL,SHAR,SHLL,SHLR,ROTL,ROTR,ROTXL,ROTXR
位操作	BSET,BCLR,BNOT,BTST,BAND,BIAND,BOR,BIOR,BXOR,BIXOR,BLD,BILD,BST,BIST
分叉	Bcc*,JMP,BSR,JSR,RTS
系统控制	TRAPA,RTE,SLEEP,LDC,STC,ANDC,ORC,XORC,NOP
块传递	EEPMOV

注:Bcc是条件分叉命令的总称。

表2.6 H8/300H CPU的所有命令(摘自:H8微控制器入门,堀桂太郎著,2003)

2.3.3 H8的程序开发环境

图2.63给出了用H8进行程序开发的流程,它基本上与PIC的开发过程相同。但是,H8无法得到像PIC那样免费下载的开发环境MPLAB,因此,花费一些投资是必要的。

程序开发工具是由Renesas Technology提供的**HEW环境**,但若个人购买,价格太高。想便宜一些,就只有使用秋月电子商务出售的软件(约2000日元左右)。

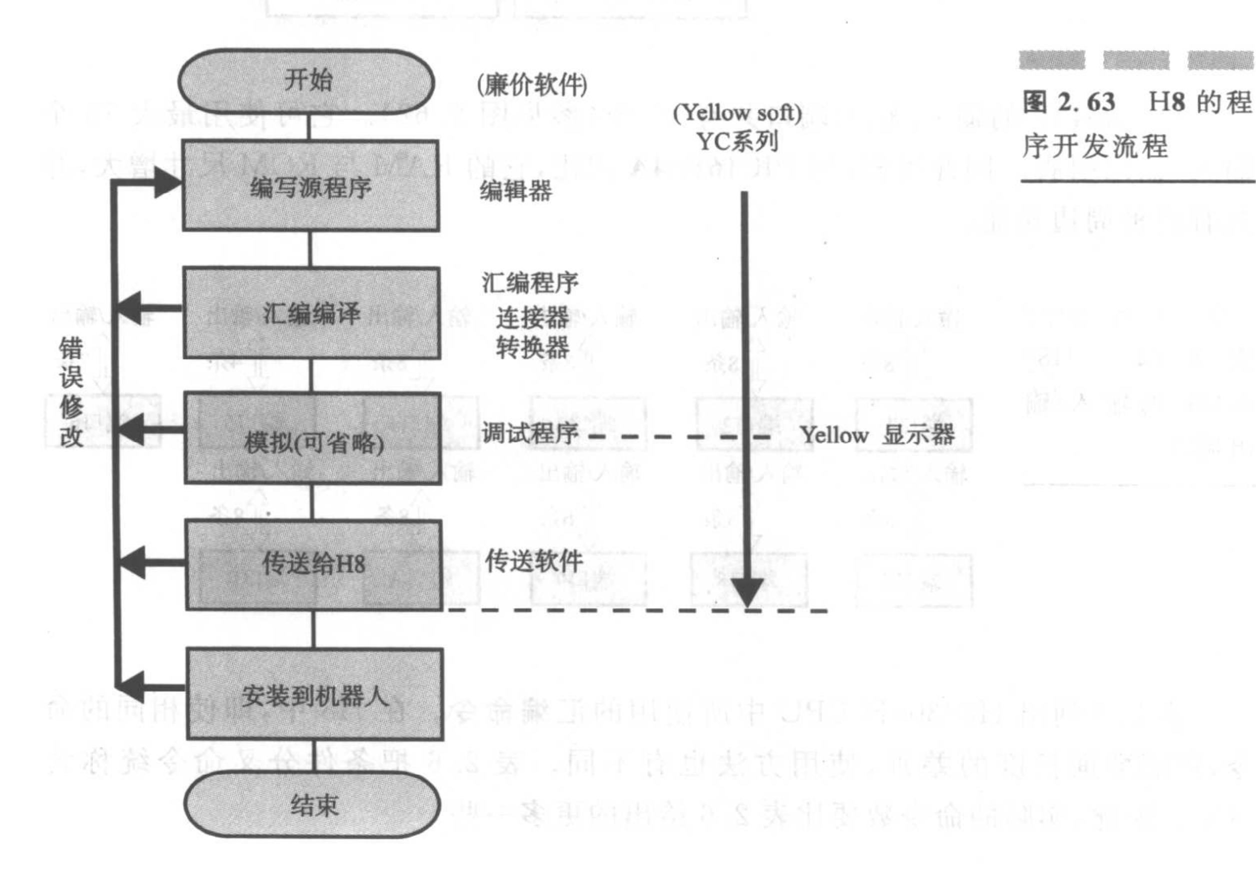


图2.63 H8的程序开发流程

由 Yellow-soft 公司开发的 **YC 系列 C 编译器/汇编程序**（约 28000 日元左右）是一个包含了汇编程序以及 C 编译器的综合开发环境，能完成从程序编写到向 H8 传送的多种功能（参见图 2.64）。

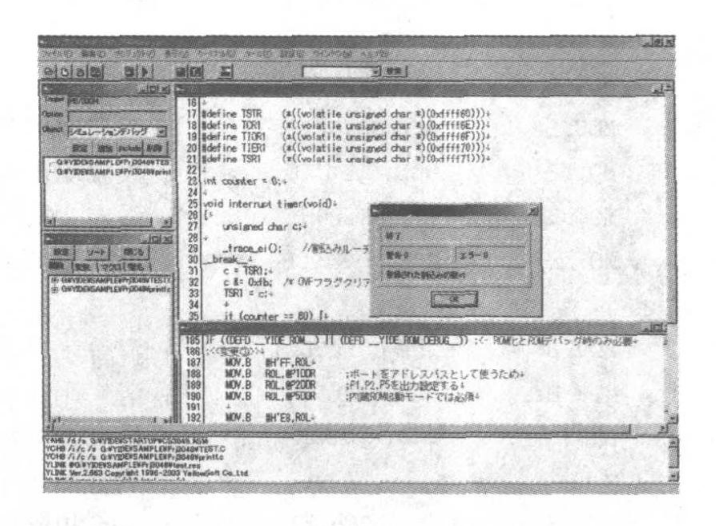
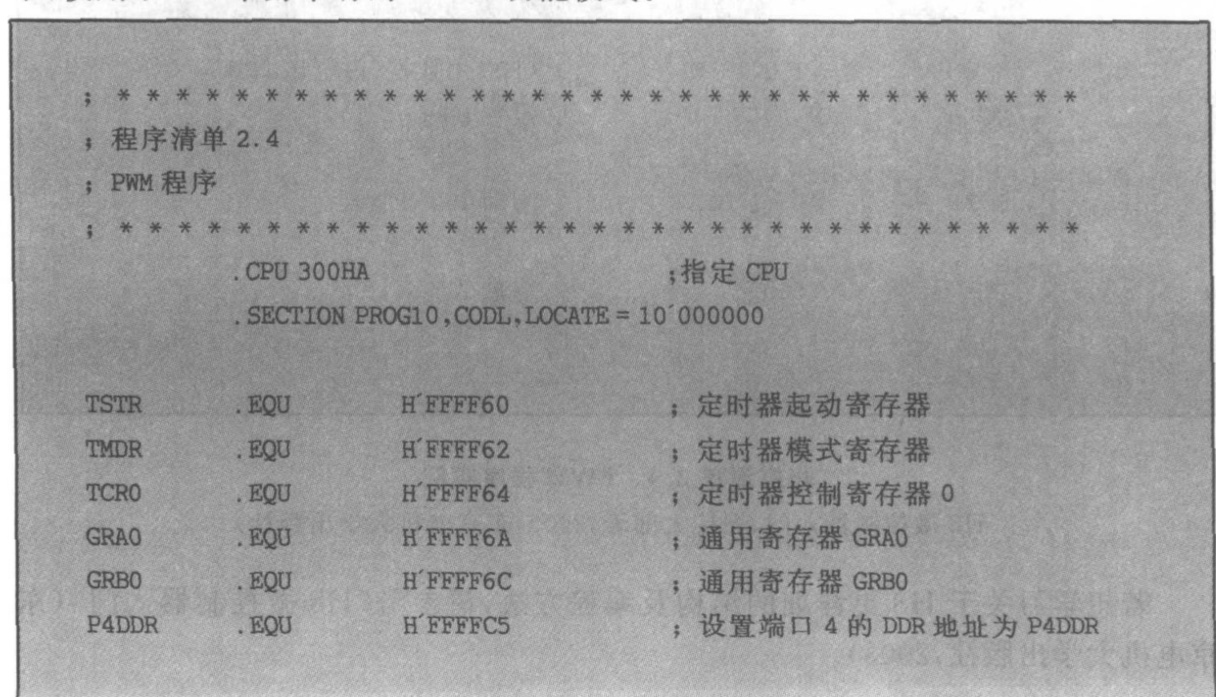


图 2.64 YC 系列 C 编译、汇编器

除此之外，为 H8 提供程序开发环境的还有 GCC，它是一种能免费使用的 C 编译器。

下面的例子，即程序清单 2.4 给出了采用 **PWM** 实现直流电机速度控制程序。程序是针对 H8/3048F 编写的，**DC 电机**与端口 A 的 TIOCA0 引脚连接。与端口 4 的 P44 引脚连接的按钮开关利用 ON/OFF 来改变电机的转动速度。在这个例子中，使用了 H8 本身带有的 PWM 功能模式。



```
; ***********************************
; 程序清单 2.4
; PWM 程序
; ***********************************
          .CPU 300HA                       ;指定 CPU
          .SECTION PROG10,CODL,LOCATE = 10'000000

TSTR      .EQU        H'FFFF60             ; 定时器起动寄存器
TMDR      .EQU        H'FFFF62             ; 定时器模式寄存器
TCR0      .EQU        H'FFFF64             ; 定时器控制寄存器 0
GRA0      .EQU        H'FFFF6A             ; 通用寄存器 GRA0
GRB0      .EQU        H'FFFF6C             ; 通用寄存器 GRB0
P4DDR     .EQU        H'FFFFC5             ; 设置端口 4 的 DDR 地址为 P4DDR
```

```
P4DR      .EQU      H'FFFFC7          ; 设置端口 4 的 DR 地址为 P4DR
P4PCR     .EQU      H'FFFFDA          ; 设置端口 4 的 PCR 地址为 P4PCR

          .SECTION ROM, CODE,LOCATE = H'000100

          MOV.L     #H'FFFF00,ER7     ; 设置 SP
          MOV.B     #H'00,R0L         ; 设置输出/输出数据
          MOV.B     R0L,@P4DDR        ; 设置端口 4 为输入
          MOV.B     #H'FF,R0H         ; 设定上拉用数据
          MOV.B     R0H,@P4PCR        ; 设定端口 4 上拉有效

          MOV.B     #B'00100011,R0H   ; 清除计数器、1/8 周的数据
          MOV.B     R0H,@TCR0         ; 清除计数器,设定预分频器
          BSET      #0,@TMDR          ; 设定 PWM 模式

          MOV.W     #H'00C8,E1        ; GRA0 用数据
          MOV.W     E1,@GRA0          ; 设定 GRA0

LOOP:     BTST      #4,@P4DR          ; 检查端口 4 的开关
          BEQ       D20               ; 开关 ON 时,转向 D20
          MOV.W     #H'00C8,E1        ; GRB0 用数据(占空比 100%)
          MOV.W     E1,@GRB0          ; 设置 GRB0
          JMP       @D100             ; 转移到 D100

D20:      MOV.W     #H'0028,E1        ; GRB0 用数据(占空比 20%)
          MOV.W     E1,@GRB0          ; 设置 GRB0

D100:     BEST      #0,@TSTR          ; 起动 ITU 计数器

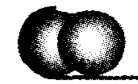
          JMP       @LOOP             ; 重复循环

          .END
```

程序清单 2.4 PWM 程序实例

(H8 微控制器入门,堀桂太郎著,2003,东京电机大学出版社)

若想学习关于 H8 更详细的结构及编程方法,请参考《H8 微控制器入门》(东京电机大学出版社,2003)。



练习题

1. 设计图 2.65 所示的 PIC 实习电路，编写各种控制程序。

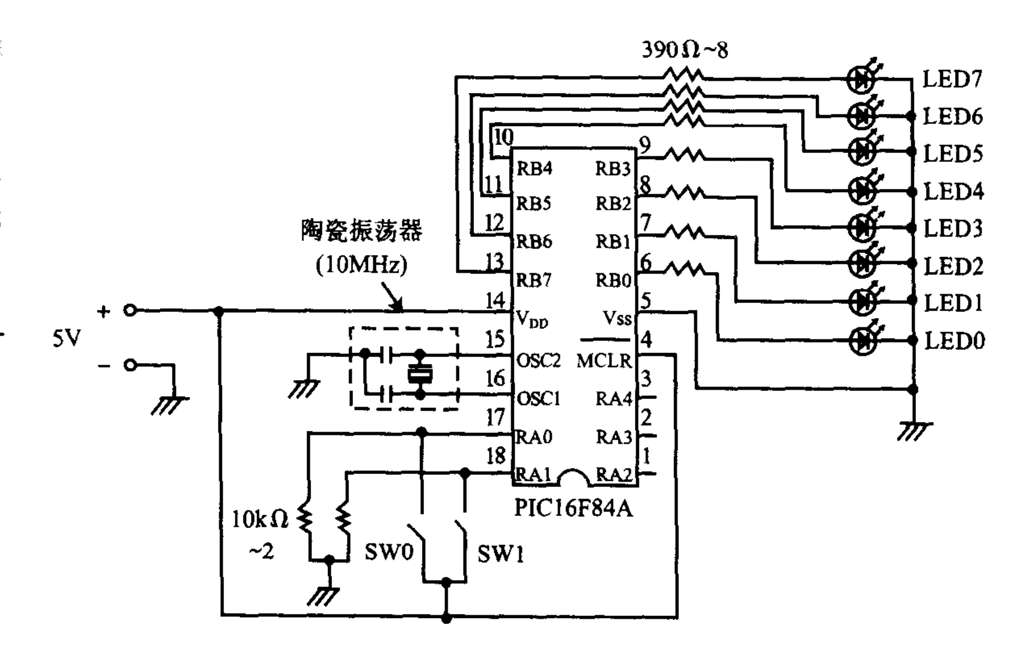
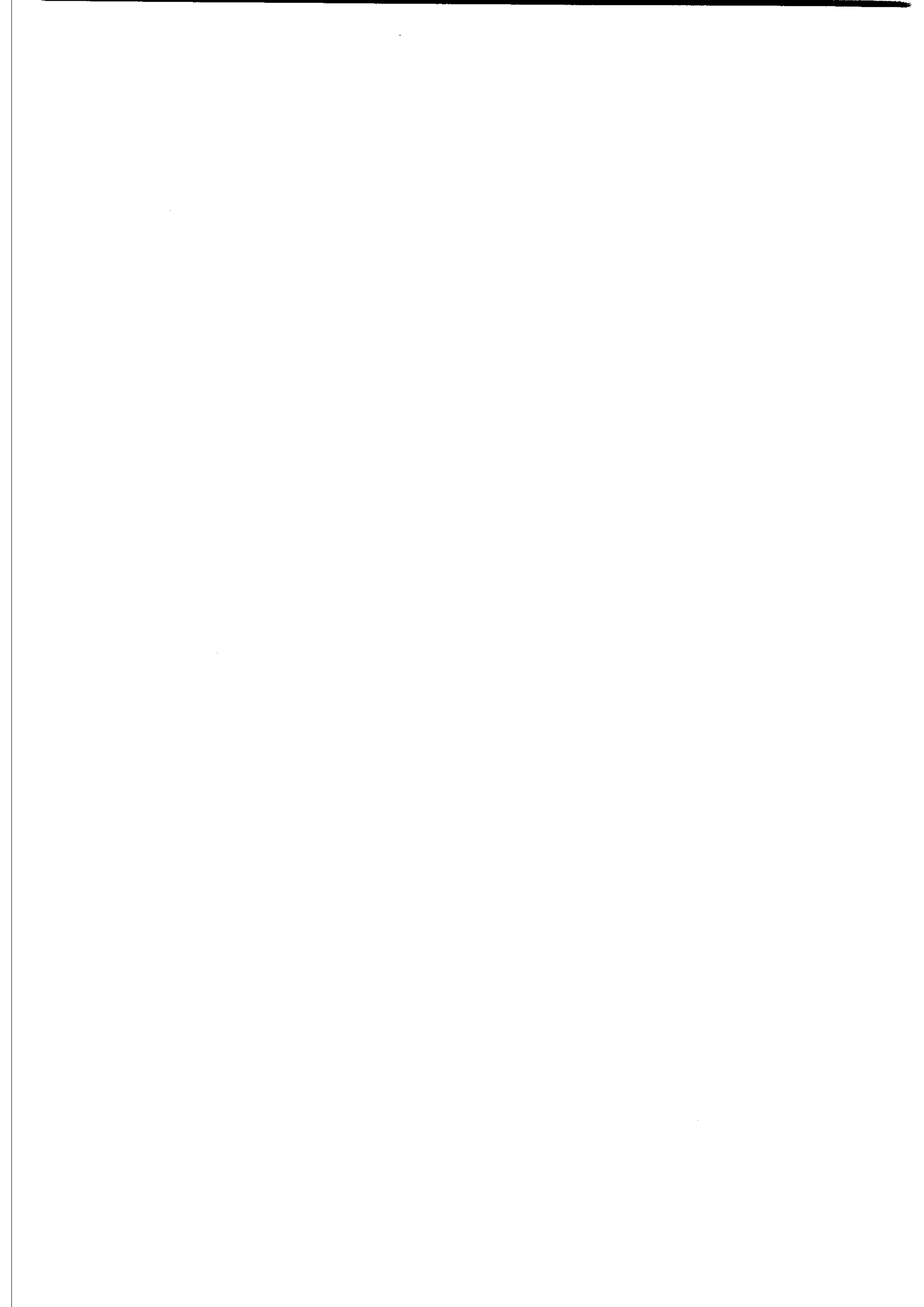


图 2.65 PIC 实习电路（摘自：图解 PIC 微控制器实习，堀桂太郎著，2003，森北出版社提供）

2. 试讨论 PIC16F84A 内部搭载的存储器。
3. PIC 是所谓的哈佛构造，请简单说明。
4. 比较采用 PIC 或 H8 进行微控制器控制时各自的优缺点。
5. 列举用 PIC 开发程序时所需要的软硬件。
6. 简单说明单片机是一种什么样的微控制器。
7. 画出与程序清单 2.4 的 PWM 程序所对应的 H8 与直流电机的接线图。



第3章

直流电机的控制

直流电机(DC电机)是日常生活中广泛使用的一个电气产品，好像人的肌肉一样，它是机器人中最重要的动力源。它常常与第5章所要学习的传感器组合起来使用，以便让机器人完成向给定位置移动或搬运某物体等动作。实现这样的动作，需要能进行转动控制和速度控制的电路，实际中通常采用微控制器和专用IC芯片。单独使用直流电机尚不能达到精确的定位控制，只有将它与旋转编码器组合起来，才能实现精确的位置控制和速度控制。

本章将学习直流电机的组成及基本的控制方法，并简单地介绍旋转编码器的组成和使用。

3.1 直流电机的基础知识

电机是我们的生活中不可或缺的动力源，常用的有交流电机和**直流电机**。机器人中作为控制对象的电机大都是靠电池供给能源而转动的直流电机。

直流电机的组成及原理

图3.1是常用的3槽(极)直流电机的构造。在绕有线圈的铁心(转子)的外周，有相同数目的定子(永久磁铁)相向包围。通过整流器的电刷转子断续地得到电流，转子与定子的永久磁铁反复相吸、排斥，结果无论它们的相互位置如何，转子始终能够沿着一定的方向回转。图3.2是组成直流电机的各部分的照片。

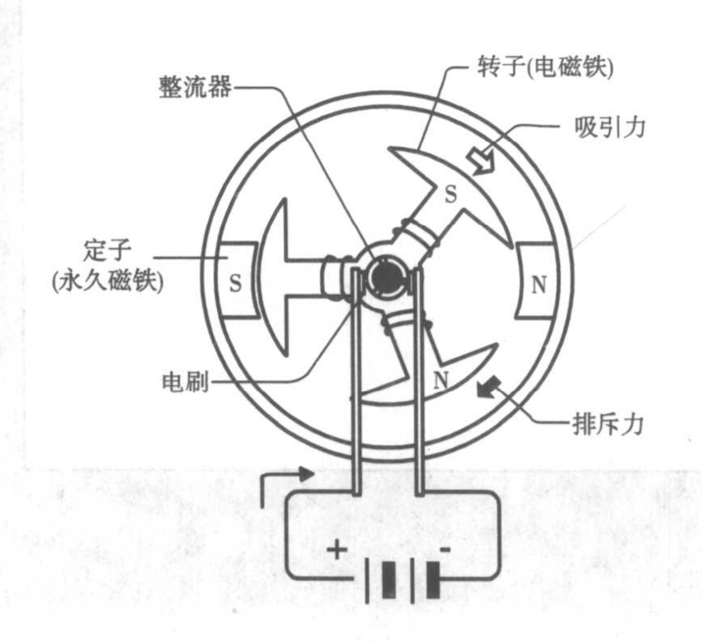


图3.1　直流电机的构造

图 3.2 直流电机的组成部件

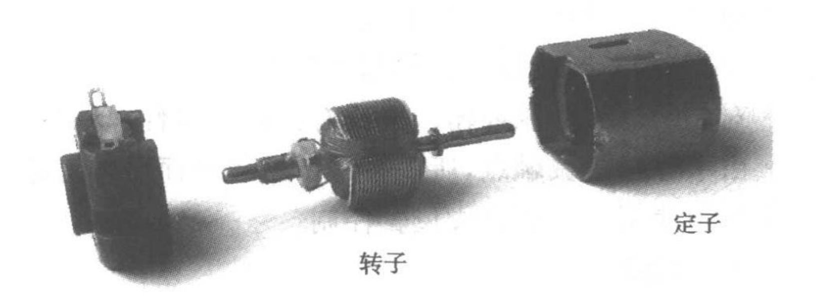
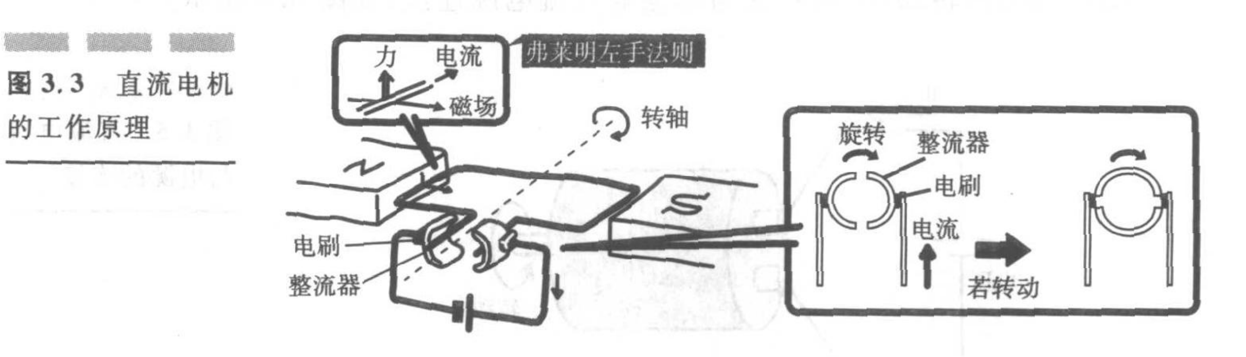


图 3.3 解释了直流电机的工作原理。在永久磁铁(N、S 极)之间的线圈有电流通过时，产生电磁力，对应于永久磁铁产生的磁场，N 极产生向上的力，S 极产生向下的力，从而使线圈动作。结果导致线圈旋转。

在线圈转动的同时，整流器也随之旋转，这样就能保证电流流向的自动切换以便维持转子向同一方向的连续转动。于是，线圈中不断地产生相同方向的旋转力。

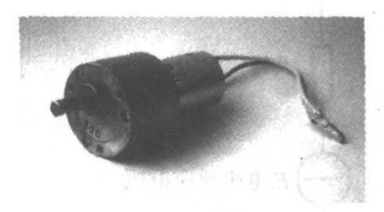
根据弗莱明(Fleming)左手法则，电机产生转矩，而电机一旦开始回转，就使线圈磁场中转动，再由弗莱明右手法则，应该有电压产生，称之为反电动势，外部所施加的电压克服反电动势，线圈中就有电流流过。

图 3.3 直流电机的工作原理

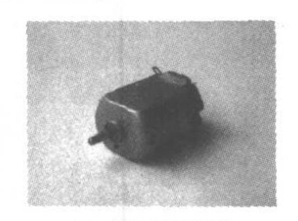


直流电机中的电刷与整流器实际上属于机械接触，由于在转轴转动的过程中它们的接触状态不断发生变化，往往产生电气噪声，这可能成为诱发 CPU 电路误动作的原因，因此为了克服噪声干扰，一般在电机电源端子上并联一个电容器。图 3.4 为直流电机的外形。

图 3.4 直流电机的外形



(a) 带减速器的电机



(b) 模型用电机

直流电机的特征可归纳如下：

- 可高速旋转，能取得大的起动转矩。
- 对应于输入电压或输入电流有良好的线性，易控制。
- 由于存在机械触点，所以容易产生噪声而影响寿命。
- 单独使用时不能完成位置控制。
- 价格低廉。

3.2 直流电机的转动控制

简单地将直流电机与直流电源或电池连接就可以实现转动。不过作为机器人的动力使用时，必须对电机进行转动的开关（ON/OFF）和转动方向等控制。本节通过对直流电机驱动方式和与微控制器连接实例的讲解，学习关于直流电机的转动控制方法。

3.2.1 直流电机的驱动电路

使直流电机转动，必须将它与电池等直流电源连接，如图3.5所示。

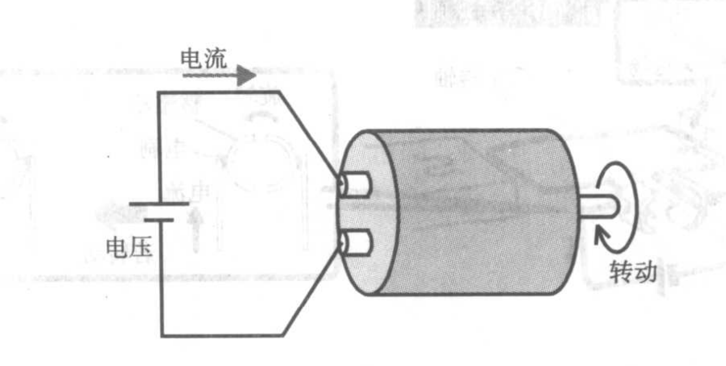


图3.5 直流电机与电源的连接

以恒定的速度转动时DC电机的等效电路如图3.6所示。这里，R_a 是转子线圈的电阻，E_c 表示转子线圈在定子磁场中转动时产生的反电动势。

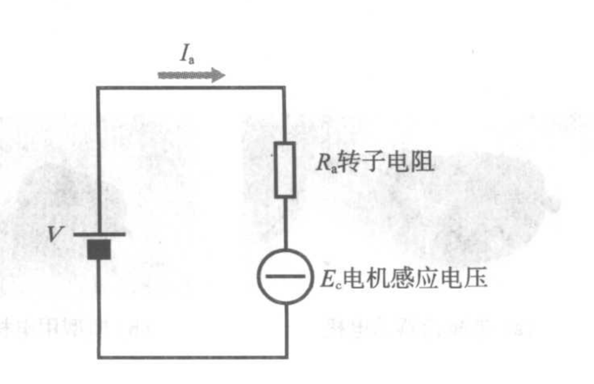
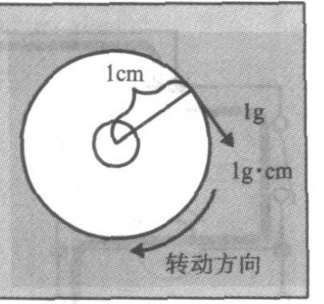


图3.6 DC电机的等效电路

若用公式表示，则有 $V=R_a I_a+E_c$(V)。这里忽略了电刷与整流子之间的接触电压。

专栏：转矩

电机产品目录中常出现转矩这样的术语，它既代表电机轴转动的驱动力，也意味着停止电机转动的制动力，它的单位是 mN·m 或 g·cm。1g·cm 转矩的意义是在离开轴中心 1cm 的地方施加 1g 大小的力使轴转动。两个单位之间的关系是：mN·m=g·cm×0.098。



通常采用半导体或继电器来驱动直流电机。继电器电路比较简单，缺点是由于利用机械触点，所以易产生噪声，反应速度也比较慢。本节以图 3.7 所示的半导体电路为主实现 DC 电机的驱动。为了完成直流电机的控制，采用的晶体管一般都是大功率型的。

图 3.7 直流电机的驱动电路

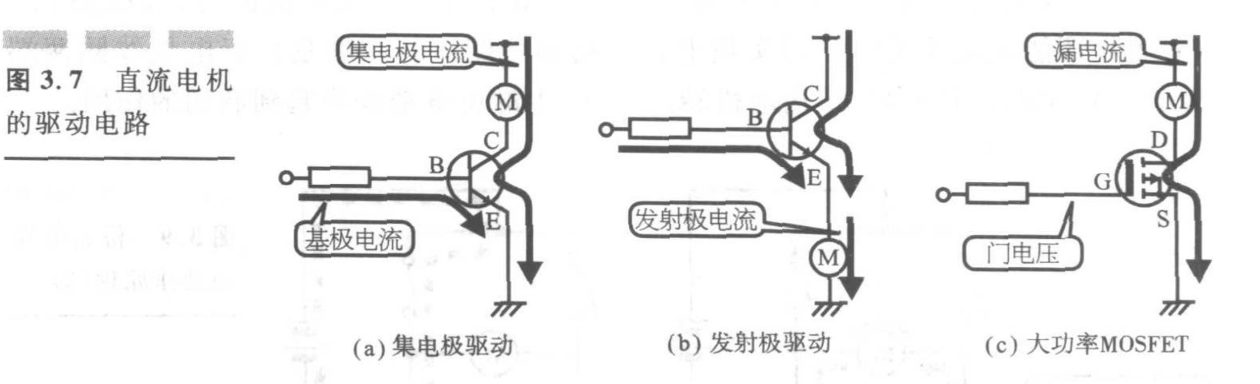


图 3.7(a)是常用的电路，三极管的集电极与直流电机连接，改变基极的电压可以控制与集电极连接的电机的电流。由于驱动力大，所以这种方式可用于电机的开关(ON/OFF)控制。

图 3.7(b)把三极管的发射极与直流电机连接，改变基极电压可以控制与发射极连接的电机的电压。由于三极管起到可变电阻的作用，所以它会造成电压降并损耗一定的电能。在低速工作的场合，电压降较大，导致电能的使用效率下降，所以常用于速度控制。

图 3.7(c)的电路与图 3.7(a)相似，但是改用大功率 MOSFET。大功率 MOSFET 是电压驱动元件，仅仅施加门电压即可动作，因此电路简单，可在高速下工作。此外，该类元件工作的热特性稳定，最近使用得越来越多。

不过上面介绍的电路只能使电机沿一个固定的方向转动，不大符合机器人的实际要求，因为它不仅需要前进，也需要后退。这时，可以改成图 3.8 所示的**桥式**

驱动电路。该电路可实现电机的正转、反转。在图 3.8 中开关可以采用三极管或 FET。

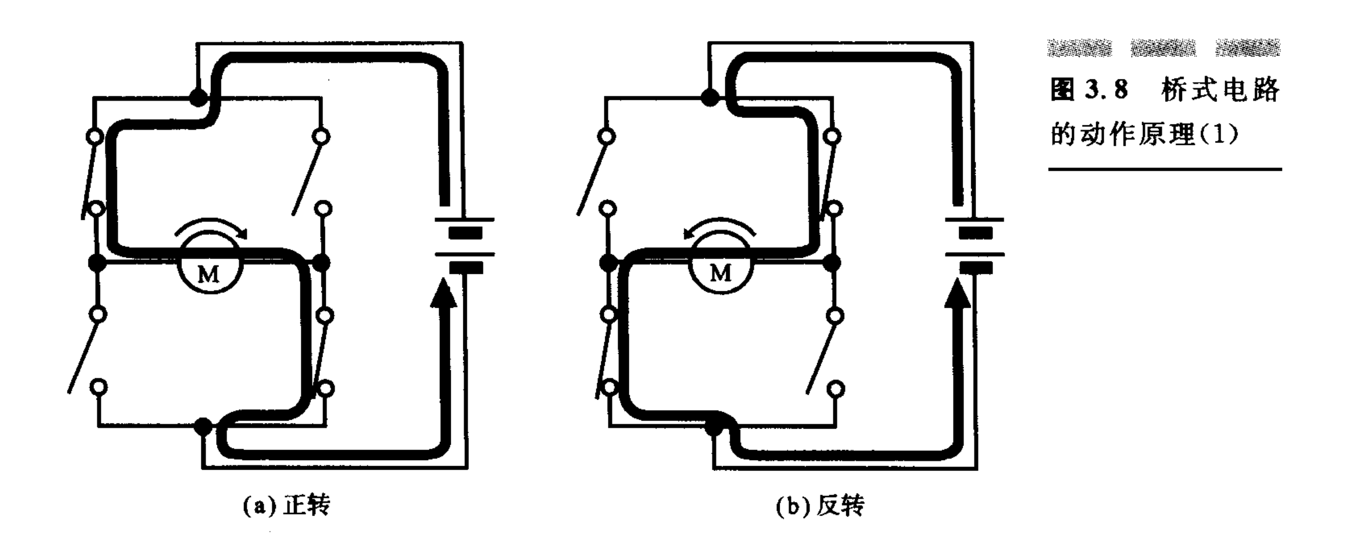


图 3.8　桥式电路的动作原理(1)

图 3.8 借助对角线方向的一对开关的开闭组合来控制电机的转向，而图 3.9(a)表示还能利用开关的开闭实现电机的制动功能。其原理是由于电机线圈两端处于短路状态，当外力企图使电机转动时，产生的反电动势将起到制动的作用。

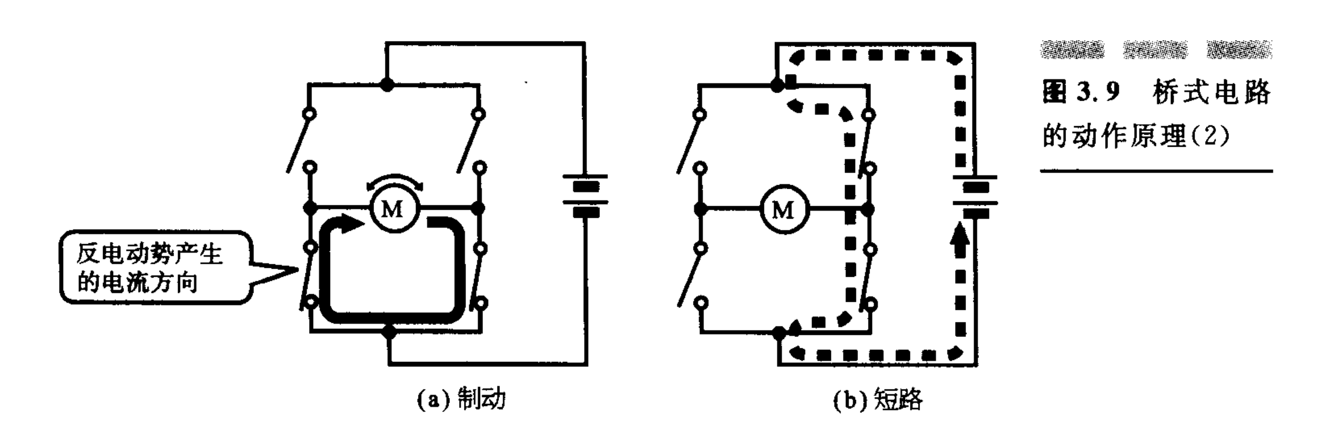


图 3.9　桥式电路的动作原理(2)

可是，在桥式电路中，当同侧开关处于图 3.9(b)所示的闭合状态时，电源实际上被短路，电路中将出现过大的电流。这时，原来担负开关任务的三极管或 FET 将被损坏，因此必须注意避免这种情况的发生。

直流电机控制的实际电路中都有防止短路的保护电路。图 3.10 是专用的电机驱动 IC，如果选用这样的驱动芯片，电路设计就变得非常简单了。

在本章的实验中，所采用的驱动芯片(IC)是东芝株式会社制造的 TA7257P。下面就它的内部电路的组成、功能以及使用方法加以说明。



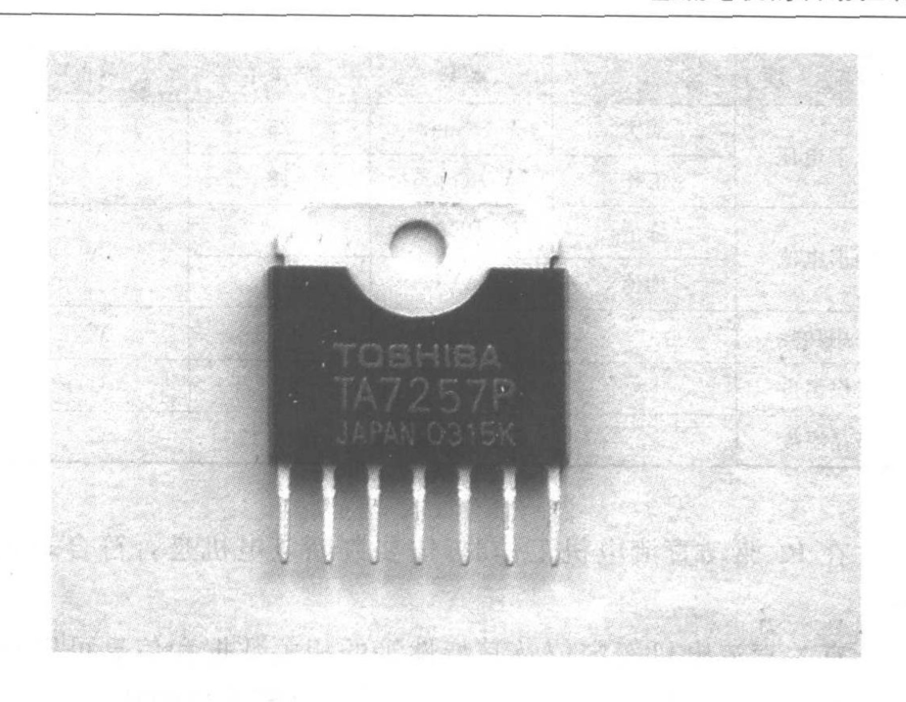
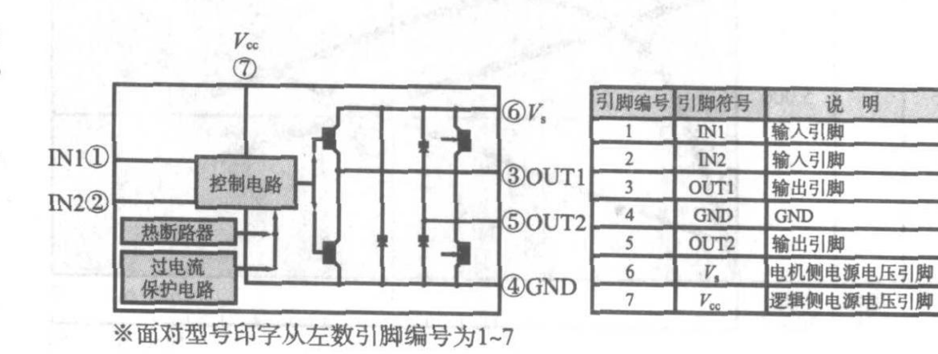


图 3.10 电机驱动的芯片 TA7257P

图 3.11 是这个芯片的结构框图。其内部正是由上面已经介绍的三极管桥式电路组成的。

TA7257P 的控制方法是如果向引脚 1、2 施加表 3.1 中列出的信号，它就会按照规定的模式动作。如置引脚 1(IN1)为 0，引脚 2(IN2)为 1，则电机正转(取决于电机端子连接的＋，－状态，也可以反转)。表 3.2 中给出了 TA7257P 的性能规格。它的工作电源电压范围 V_{cc} 是 6～18V，V_s 为 0～18V。



引脚编号	引脚符号	说 明
1	IN1	输入引脚
2	IN2	输入引脚
3	OUT1	输出引脚
4	GND	GND
5	OUT2	输出引脚
6	V_s	电机侧电源电压引脚
7	V_{cc}	逻辑侧电源电压引脚

图 3.11 TA7275P 的结构框图

表 3.1 TA7275P 的功能

IN1	IN2	OUT1	OUT2	模 式
1	1	L	L	制动
0	1	L	H	正/反转
1	0	H	L	反/正转
0	0	高阻抗		停止

表 3.2　TA7275P 的最大额定值

项　　目		符号	额定值	单位
电源电压	最大	V_{CC}(max)	25	V
	工作	V_{CC}(opr)	18	
输出电流	峰值	I_O(PEAK)	4.5	A
	均值	I_O(AVE.)	1.5	
许用功耗		P_D	12.5	W
工作温度		T_{opr}	−30～75	℃
保存温度		T_{stg}	−55～150	℃

在 IC 驱动直流电机工作时，需要判断该电机是否符合表 3.2 所给出的额定范围。

有关直流电机额定值及其特性等的相关数据和信息可以从生产厂家网页的产品样本目录中了解到。图 3.12 表示某一直流电机的特性。

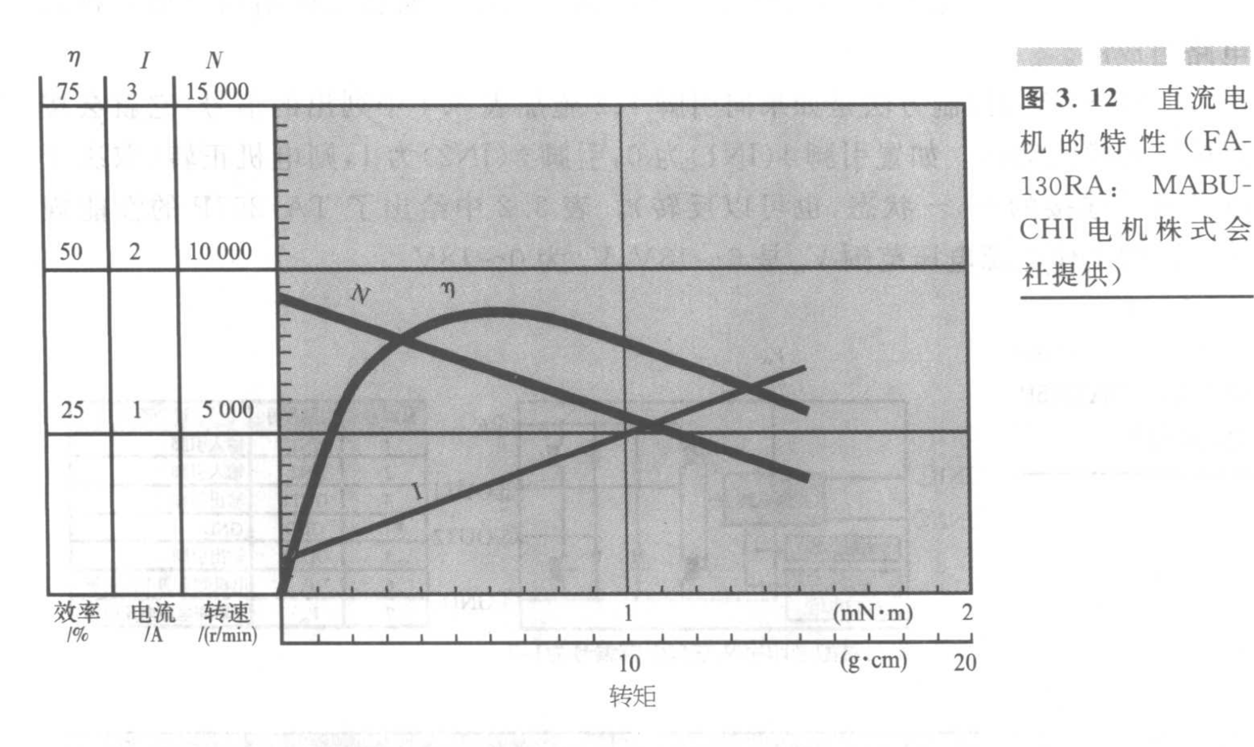


图 3.12　直流电机的特性（FA-130RA：MABUCHI 电机株式会社提供）

3.2.2　用微控制器控制直流电机转动

在这一节，打算采用第 2 章介绍的微控制器构建一个如图 3.13 所示的直流电机转动控制系统。

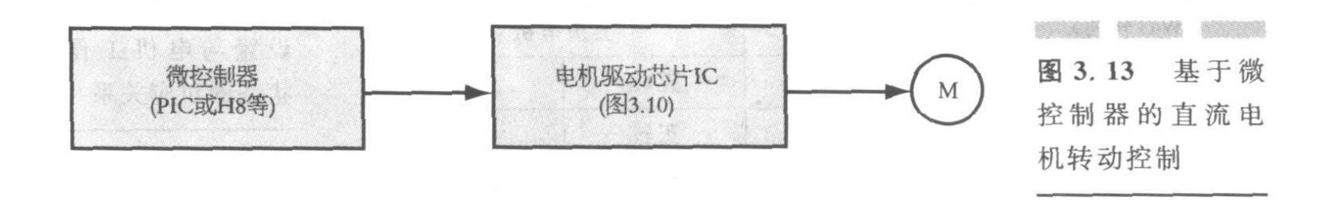


图 3.13 基于微控制器的直流电机转动控制

对微控制器进行编程，根据程序改变电机驱动芯片 IC 的控制信号就能按照给定的要求进行转动控制。

下面就以 PIC 微控器为例来做直流电机的转动控制实验。

图 3.14 中介绍了一个用第 2 章的微控制器 PIC16F84A 实现直流电机控制的具体电路。

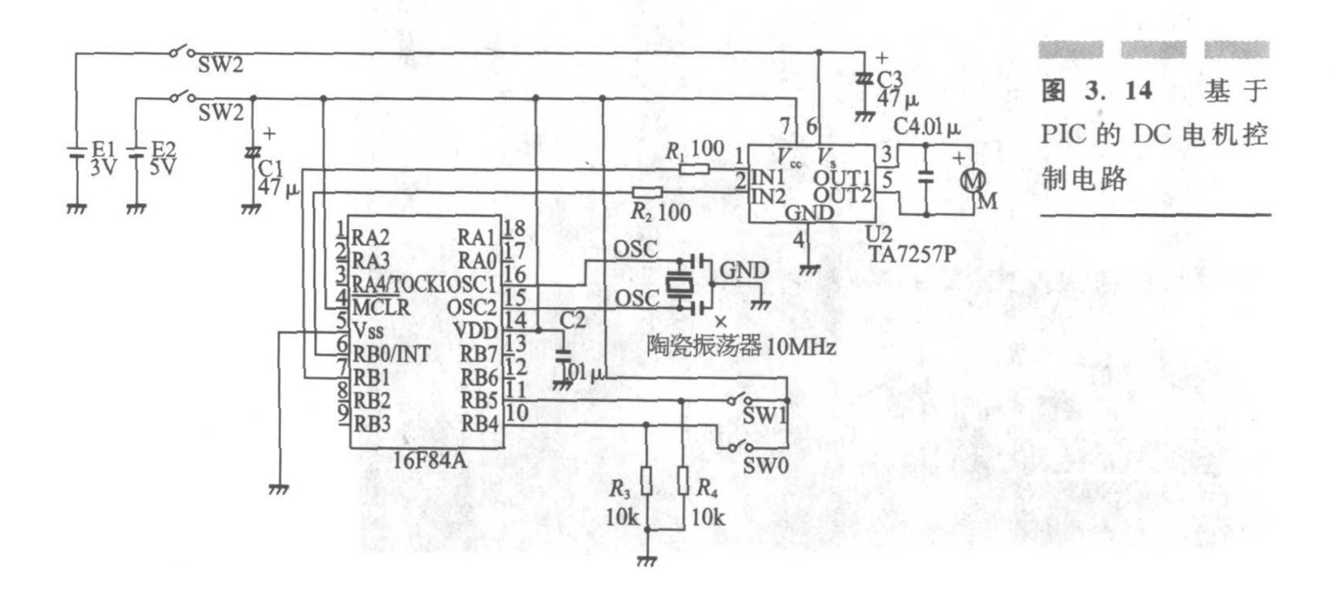


图 3.14 基于 PIC 的 DC 电机控制电路

设置 PIC 的端口 B 的 2 位(RB0～RB1)为 IC 控制的输出，与电机驱动芯片 IC 连接。另外，再设置端口 B 的另外 2 位(RB4～RB5)作为输入，与设置电机转向和转动状态的开关连接(实际的程序中，上 4 位与下 4 位分别设置为输入、输出)。整个控制过程是先读取开关的输入，判断控制的内容，然后在 2 位输出端子上输出相应的信号让 DC 电机执行。

表 3.3 给出了与输入开关相对应的电机工作状态。具体说，SW1(RB5)对应正/反转，SW0(RB5)对应连续/断续的切换。

直流电机与电源是两个不同的器件。电机驱动 IC 即 TA7257P 电源的工作电压 V_{cc}的范围为 6～18V。图 3.15 是直流电机转动控制实验的现场照片。

表 3.3　输入开关设置与电机工作状态的对应关系

输　入		输　出	
SW1(转向)	SW0(连续性)	直流电机	
OFF(0)	OFF(0)	正转	连续
OFF(0)	ON(1)	正转	间歇
ON(1)	OFF(0)	反转	连续
ON(1)	ON(1)	反转	间歇

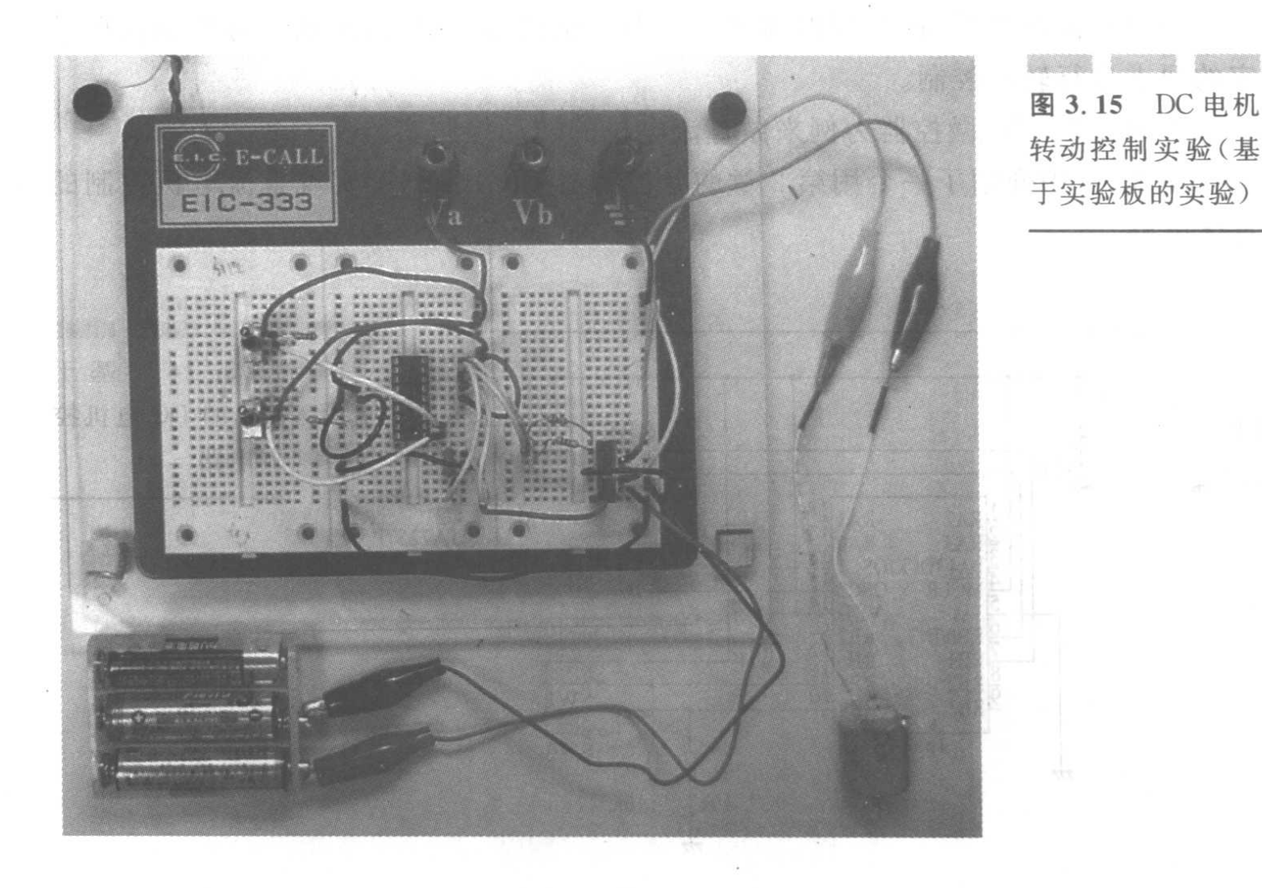


图 3.15　DC 电机转动控制实验(基于实验板的实验)

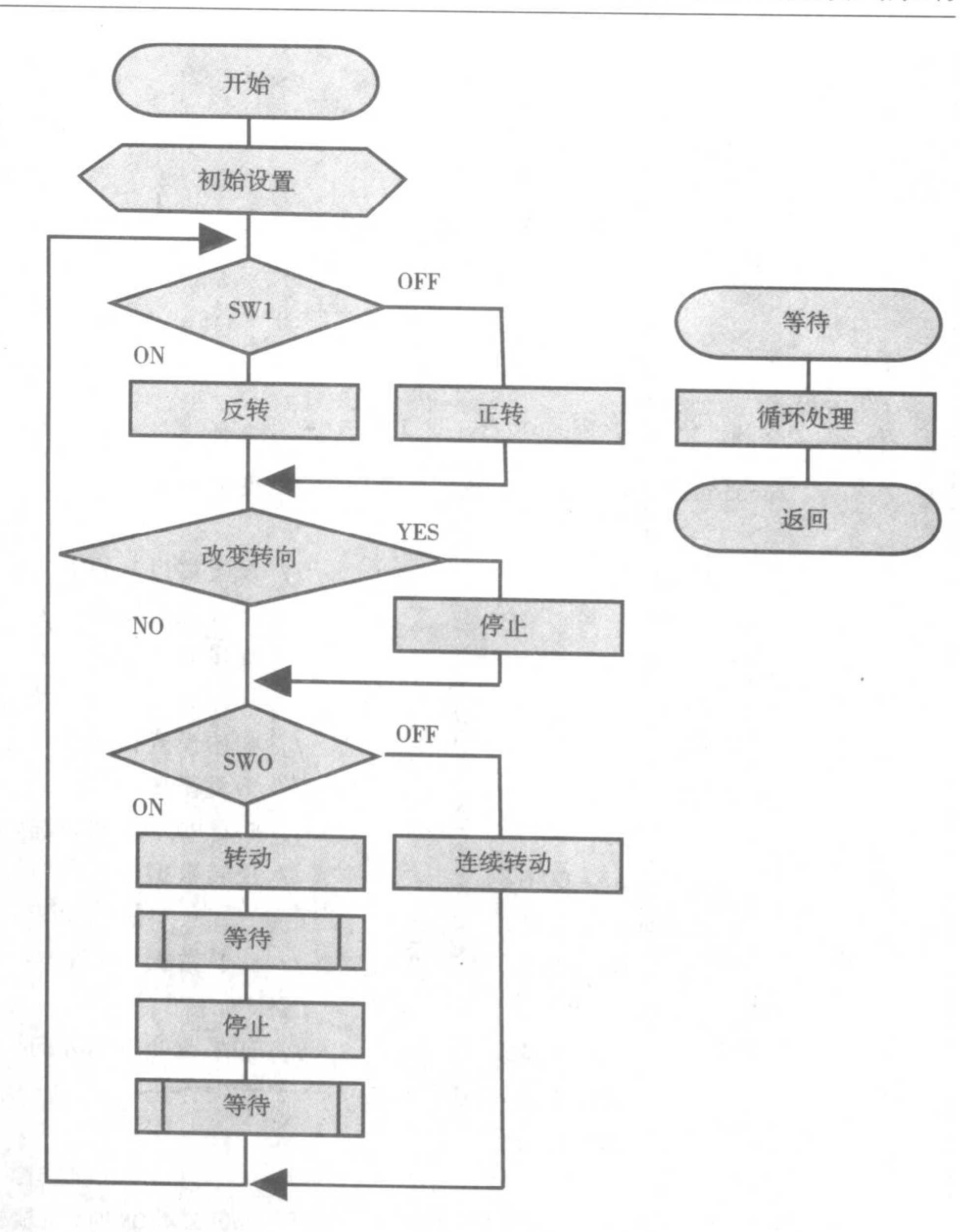
实验在实验板上完成,其方便之处在于不需要焊接。受控的 MABUCHI 电机的最大额定电压为 3V,陶瓷振荡器为 10MHz。

基于该实验平台可以进行各种电机的控制实验。例如,将开关替换成第 5 章"传感器控制"中的光电判读器,这样的 2 个受控电机就可以驱动在第 7 章将要制作的巡线小车。

图 3.16 为直流电机的控制流程框图。对应的直流电机控制源程序(C 语言)在程序清单 3.1 中给出。

在程序清单 3.1 中,通过时间等待函数(wait)的 for 语句进行循环处理,产生等待时间的循环。给定不同的变量值,即可改变等待的时间。

图 3.16 直流电机控制框图




```
// 程序清单 3.1
// 直流电机正反转控制程序
#incclude<pic.h>                              // 头文件
#define STOP            0x00                  // 停止
#define CW              0x01                  // 正转
#define CCW             0x02                  // 反转
#define BREAK           0x03                  // 制动

_ _CONFIG(0xFFF6);
// 时间等待函数
wait(unsigned inti)
```

```
{
        unsigned int j,k;
        for(j = 0;j<i;j + + ){
            for(k = 0;k<30000;k + + ){
            }
        }
}
// 主程序
main()
{       unsigned char sw, swold;            // 变量定义

        TRISB = 0xF0;                       // 设置端口 B 的上 4 位为输入,
                                            // 下 4 位为输出
        PORIB = 0x00;                       // 端口 B 清零

        sw = CW;                            // 初始设置,正转
        while(1){                           // 重复循环
            if(RB5 = = 1)                   // SW1(RB5)ON 时,反转
                sw = CCW;                   // 反转数据
            else
                sw = CW;                    // 正转数据

            if(sw ! = swold){               // 如果改变转动方向
                PORTB = STOP;               // 转动停止
                wait(1);
            }
            if(RB4 = = 1){                  // SW0(RB4)ON 时,间歇转动
                PORTB = sw;                 // SW→输出到端口 B
                wait(5);                    // 时间等待
                PORTB = BREAK;              // 制动
                wait(5);                    // 时间等待
            }else                           // 连续转动,除非 SW0(RB4)ON
                PORTB = sw;                 // SW→输出到端口 B
            swold = sw;                     // 保存旧的数据
                    }
 }
```

程序清单 3.1　直流电机的控制程序

程序执行中，等待时间的长短随 PIC 时钟频率的不同而变化，若频率为 10MHz，则等待的时间大约为 0.5 秒左右。直流电机控制命令的形式有如 PORTB＝sw，改换变量 sw，就可以控制 TA7257P 的转向。借助＃define 定义语句，分别给 CW(0x01)、CCW(0x02)、BREAK(0x03)、STOP(0x00)赋值，便可以输出表 3.1 的数据。程序清单中的 if(sw!＝swold)与 PORTB＝STOP 语句部分的作用是保证在执行 wait(1)语句进入等待中万一发生停止状态，再切换转动方向时不会造成 TA7259P 的损坏。

3.3 直流电机的速度控制

用开关(ON/OFF)控制直流电机，只能控制其转动或停止。实际应用的场合，根据任务的要求，有时需要控制直流电机的速度。

两种典型的控制电机速度的方法是改变施加在电机上的电压以及 PWM(脉宽调制)。

第一种方法，如果使用的是三极管，那么控制电路(发射极·跟踪电路)基本上与电机的开关控制差不多(参见图 3.17)。与控制电压 V_{cont} 相比，施加在电机上的端电压 V_M 仅仅低出 V_{BE}(三极管基极与发射极之间的电压降)。由于 V_{BE} 基本上不变，所以 V_{cont} 的变化也就是 V_M 的变化。

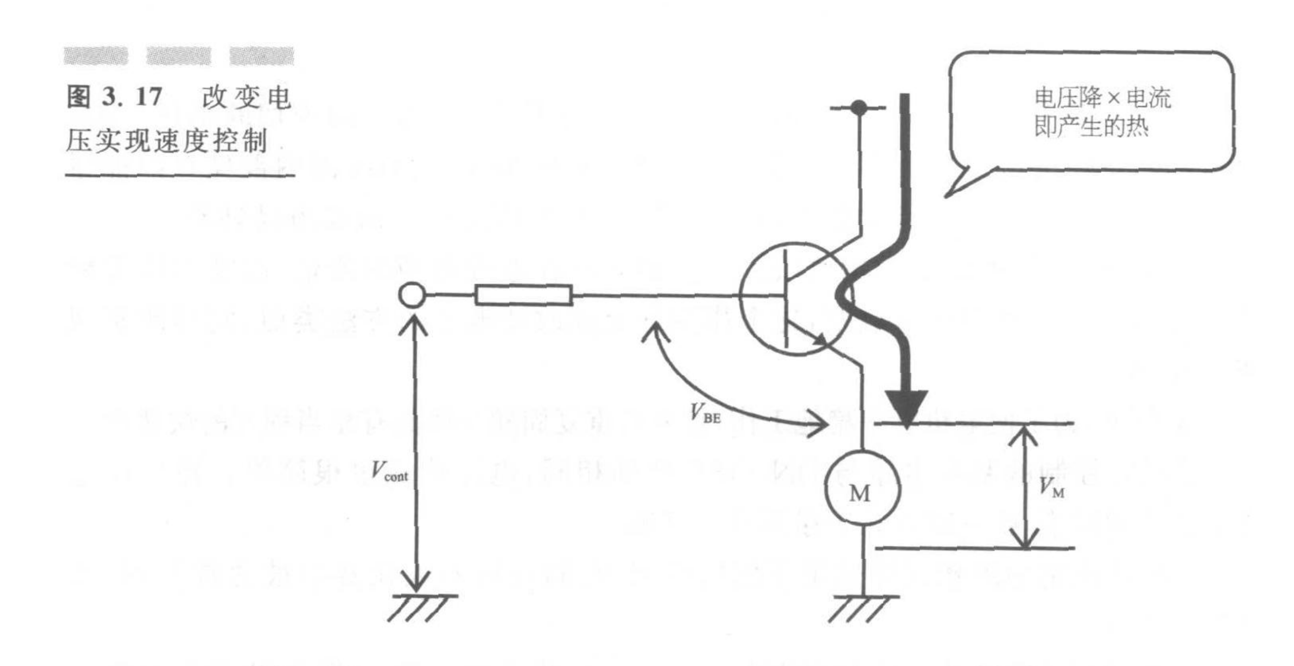


图 3.17 改变电压实现速度控制

图 3.17 的电路中，电源电压在三极管的集电极和发射极之间产生电压降，会造成集电极发热加剧，因此需要对三极管采取散热措施。特别地，在低电压、低速

转动区段，三极管中的电压降低很大，电源效率会显著降低。

第二种方法，即 PWM 控制就是给 DC 电机输入高速的开关脉冲信号，通过改变脉冲信号开关的比例，达到速度控制的效果，如图 3.18 所示。

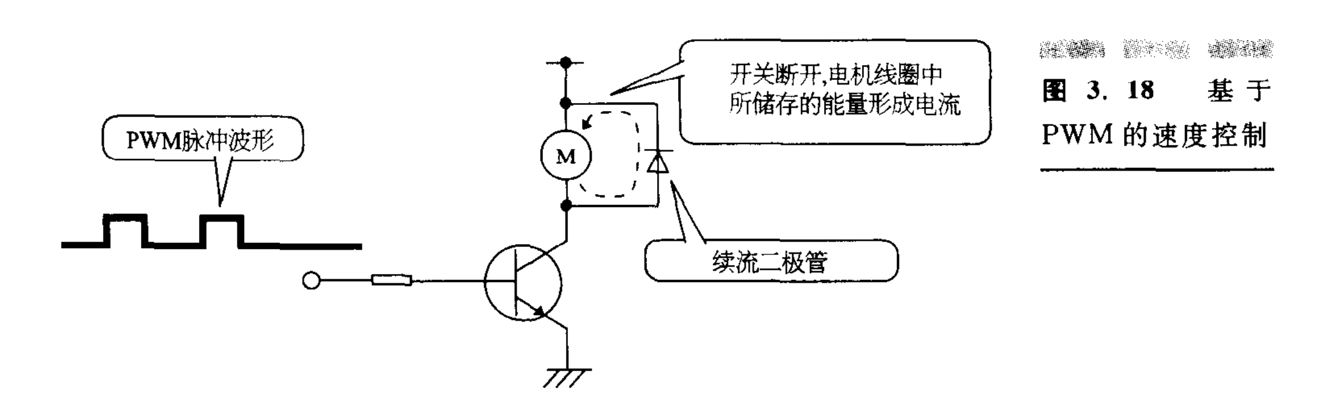


图 3.18 基于 PWM 的速度控制

施加给电机的电压呈图 3.19 所示的脉冲波形，它的占空比，即信号中 ON 和 OFF 的比例也是可以改变的。

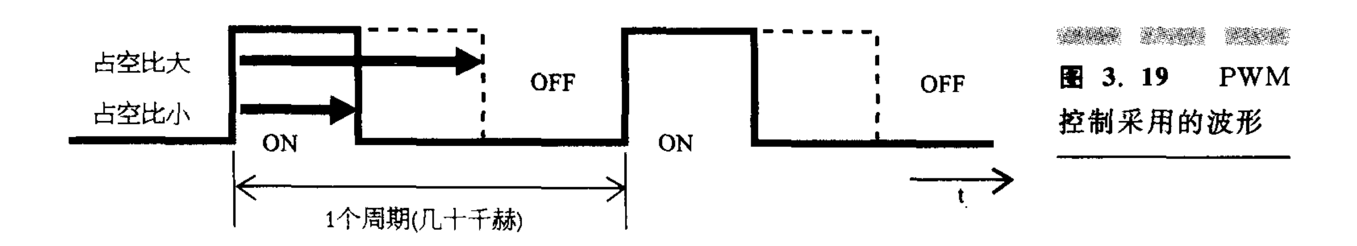


图 3.19 PWM 控制采用的波形

简单的解释是，由于在 ON 的时间内施加电压，OFF 的时间内切断电压，因此电机的转动将是断续的。不过在脉冲波形的 OFF 区段，电机线圈内部储存的能量能够产生沿续流二极管流动的电流（参见图 3.18），因此得以继续维持转动。

由此可知，脉冲波形的比例（占空比）和储存在电线圈中的能量，以及二极管释放的能量三者均可以任意改变，这个作用与上面改变电压的方法类似，同样能实现速度控制。

实际上，为了使电机能平稳地工作，脉冲的重复周期应该具有相当程度的快速性。

PWM 控制的基本电路与 ON/OFF 控制相同，电路构成也很简单。施加在电机上的 PWM 信号一般为几千赫至几十千赫。

PWM 控制的思想同样适用于舵机控制，它们在机器人比赛中被常常用到（参见图 3.20）。

在电机的 PWM 速度控制方法中还可以区分若干种方法，如将脉冲发生电路与电机驱动电路组合、PWM 控制专用芯片的硬件控制、以及借助软件产生脉冲等等。至于微控制器 PIC，也有多种型号的产品，甚至有的型号已经将 PWM 功能内置于芯

片中(如 PIC16F628、PIC16C73A 等),对应不同的应用场合,应该进行合理的选择。

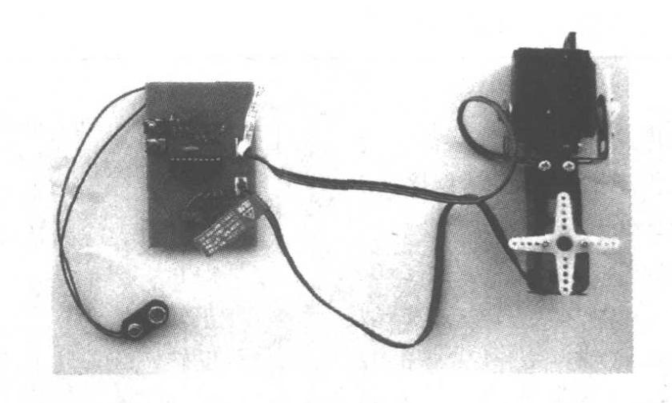


图 3.20 基于PIC的舵机控制实验

下面,就通过软件用 PIC 进行 PWM 控制的实验。

电路组成

可以直接使用图 3.14 的电路。在电机的驱动芯片 TA7257P 中,已经内置了吸收反电动势的二极管,因此,它不需要外置二极管。

图 3.21 是直流电机的速度控制流程图,程序清单 3.2 是与之对应的直流电机速度控制源程序,解读后会发现它与电机的转动控制基本相同。

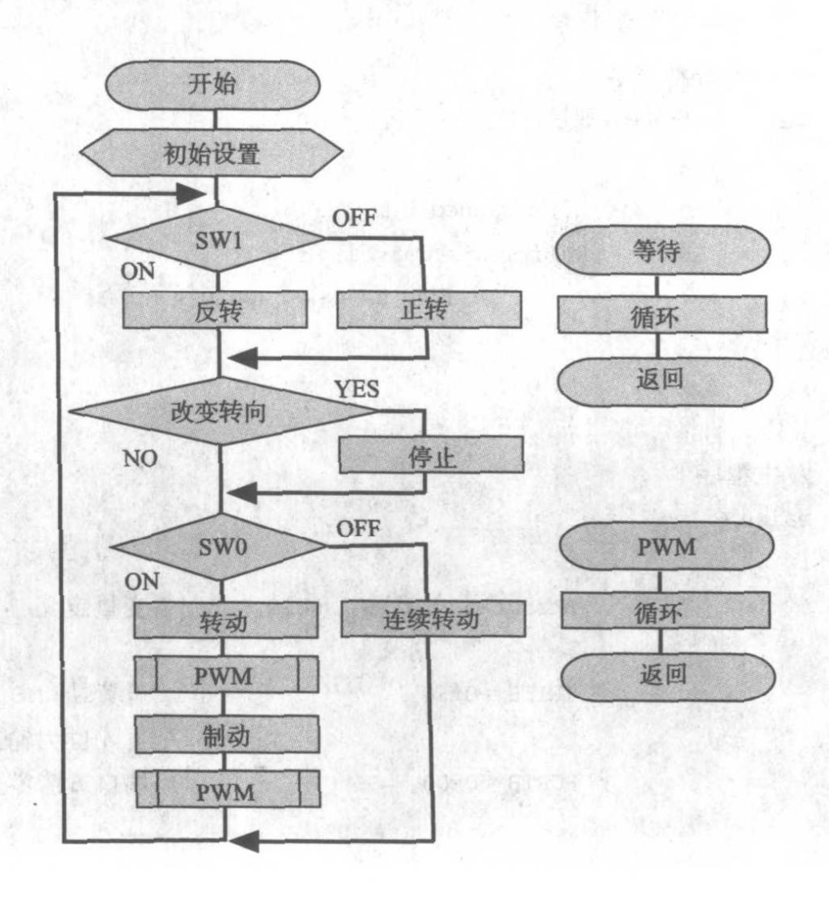


图 3.21 直流电机速度控制流程图

为此，需要向程序清单3.1的间歇转动循环周期追加快速循环处理。由于用PWM控制转速时，变化随受控电机的种类和规格有所不同，所以必须根据电机选择相适应的占空比和脉冲频率。

```
// 程序清单3.2
// 直流电机PWM控制程序
#include <pic.h>                        // 头文件
#define STOP            0x00            // 停止
#define CW              0x01            // 正转
#define CCW             0x02            // 反转
#define BREAK           0x03            // 制动

__CONFIG(0xFFF6);

//PWM时间函数
wait PWM(unsigned int x)——追加的函数
{
            unsigned int i;
            for(i=0;i<x;i++);
}
//时间等待函数
wait(unsigned int i)
{
            unsigned int j,k;
            for (j=0;j<i;j++){
                for (k=0;k<30000;k++){
                }
            }
}
//主程序
main()
{
          unsigned char sw, swold;      // 变量定义

          TRISB=0xF0;                   // 设置端口B上4位为输入，
                                        // 下4位为输出
          PORTB=0x00;                   // 端口B清零
```

```
            sw = CW;                                  // 初始设置正转
            while(1){                                 // 重复循环

              if(RB5 = = 1)                           // SW1(RB5)ON 时,反转
                    sw = CCW;                         // 反转数据
            else
                    sw = CW;                          // 正转数据
            if(sw!  = swold){                         // 若改变转动方向
                  PORTB = STOP;                       // 制动,停止转动
                  wait(1);
            }
            if(RB4 = = 1){                            // SW2(RB4)ON 时,PWM
                  PORTB = sw;                         // SW→输出端口 B
                  wait PWM(500);                      // 0.5s
                  PORTB = STOP;                       // 停止
                  waitPWM(200);
            }else
                  PORTB = sw;                         // SW→输出端口 B
            swold = sw;                               保存原数据
            }
    }
```

程序清单 3.2　速度控制程序

对于模型制作用的一般直流电机,如果采用高频 PWM 控制,一般出现的现象是当占空比例值在 0～80%之间时动力输出可能不足,反之,若超过 90%,动力输出又可能急剧地上升。

3.4 旋转编码器的使用方法

旋转编码器是测量电机等旋转运动转角的常用传感器,在机器人中也用它来实现转角和速度的控制。下面,就它的工作原理及使用方法做一些简单的介绍。

旋转编码器的工作原理

旋转编码器由光电判读器和遮光板两大部分组成,而光电判读器是包括 LED 发光元件和光电三极管一类受光元件的组件。旋转编码器可分为两大类:图 3.22 为输出脉冲信号的增量式,图 3.23 为输出数字信号的绝对式。

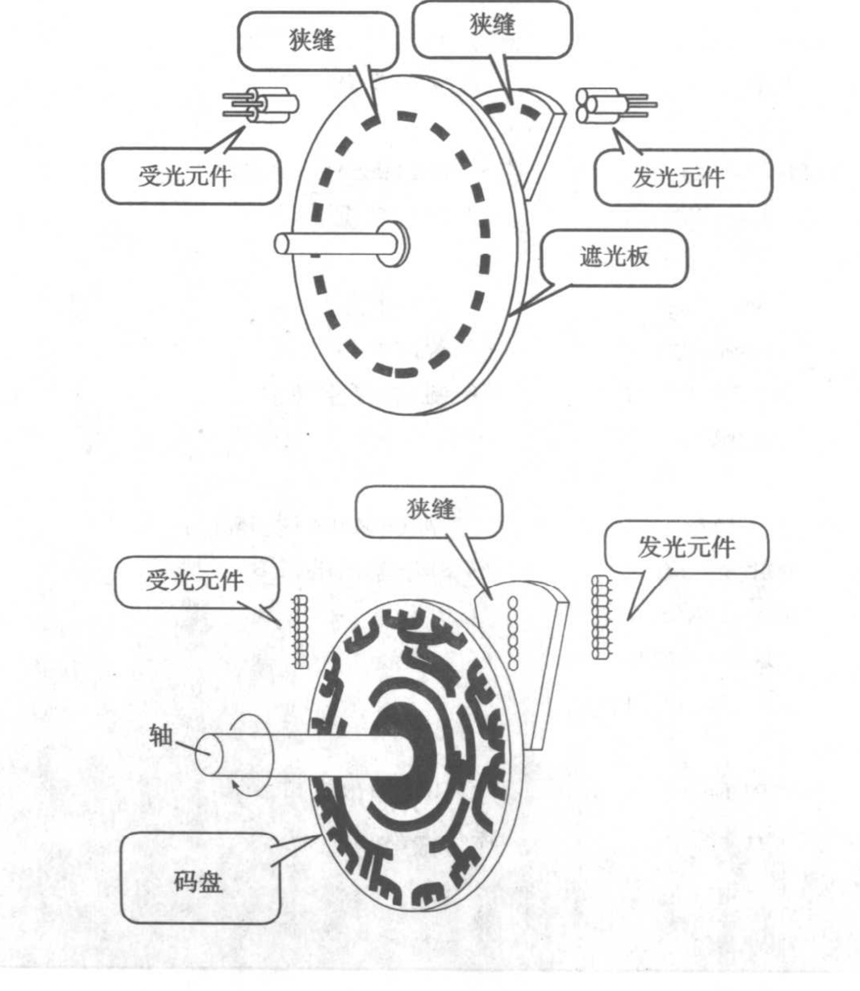


图 3.22　旋转编码器的结构（增益式）

图 3.23　旋转编码器的结构（绝对式，欧姆龙公司提供）

无论何种类型的编码器，其原理是从发光元件向受光元件发射光线，圆盘状遮光板的作用是遮挡光线。遮光板上刻有狭缝，转动时，发光元件的光必须经过狭缝才能到达受光元件。

由此可知，累计受光元件的输出信号（脉冲信号）便能检测出转动的角度。进一步，通过分析输出信号的变化率，还能得知转动速度。图 3.24 给出了旋转编码器的外形。

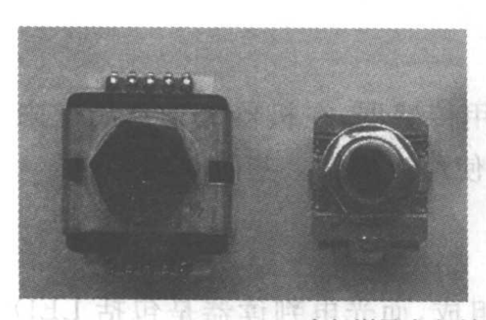


图 3.24　旋转编码器的外形

* 左起增量式(机械式、光学式),绝对式

图 3.25 表示，增量式编码器一般输出三个信号：A 相、B 相、Z 相。Z 相是确定其基准位置的输出，通常，对应于一个转动输出一个脉冲。至于 A 相和 B 相，随转动方向的改变总有其中一个信号脉冲先达到上升沿，此时只要再检测到另一信号的脉冲波形边缘便可知转动的方向。

图 3.25 增量式编码器的输出波形

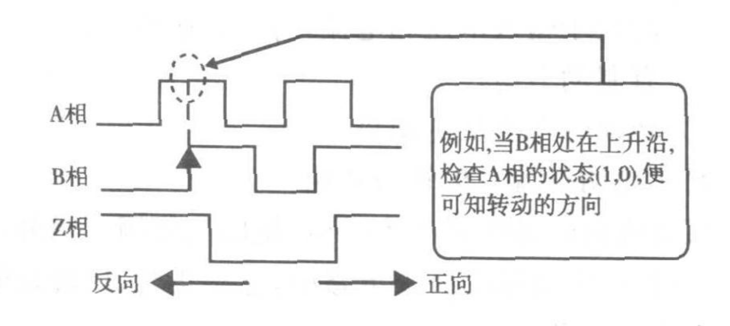


图 3.26 表示用旋转编码器组成的转角控制系统。最基本的方法是将编码器连接到电机的输出端直接检测电机的转角。图 3.27 介绍了多关节机器人角度控制的例子，改将编码器安装在可动轴部分，直接检测出轴的转角。另外，旋转编码器也用在第 8 章的微型机器鼠大赛的微型鼠动作控制，以及 ROBO-ONE 等双足步行机器人的关节运动测量等场合。

图 3.26 基于旋转编码器的角度控制系统

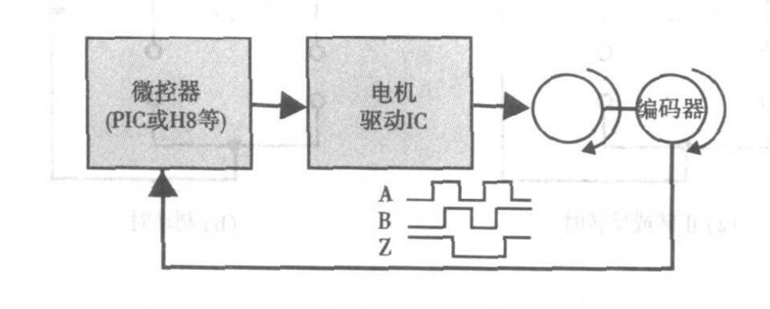
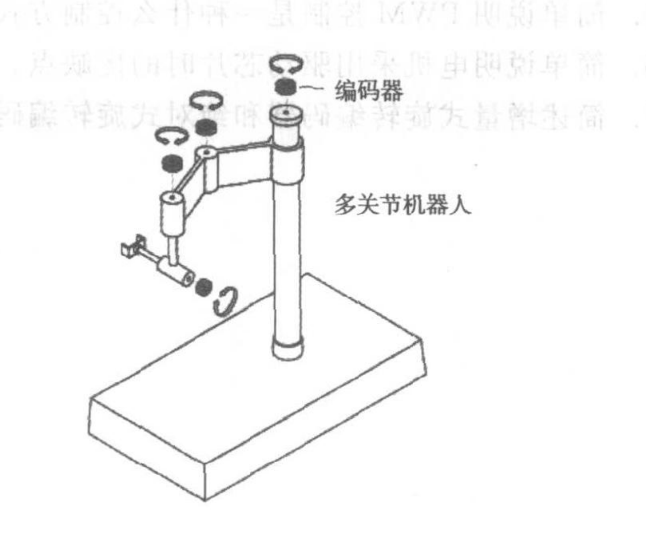


图 3.27 旋转编码器的用例（MT-engineering 公司提供）



练习题

1. 在下面①～④的表述中，哪一种说法与直流电机的特点不符合。

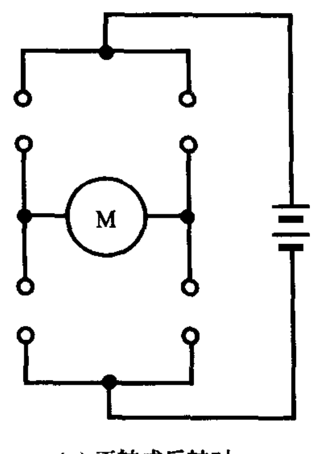
   ① 可以在高速下转动，获得大的起动力矩。

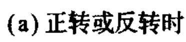
   ② 对应于输入电压和电流，有良好的线性。

   ③ 有制动力矩。

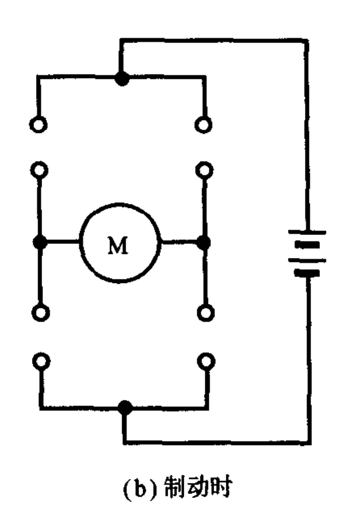
   ④ 转动时产生反电动势。

2. 简单说明续流二极管的功能。
3. 直流电机转动时会产生噪声，说明其原因。另外，如何才能防止噪声。
4. 在图 3.28 所示的桥式电路中，标出正转、反转及制动时开关的各种状态与电流的流动方向。





(a) 正转或反转时



(b) 制动时

图 3.28　桥式电路

5. 简单说明 PWM 控制是一种什么控制方式。
6. 简单说明电机采用驱动芯片时的优缺点。
7. 简述增量式旋转编码器和绝对式旋转编码器的区别。

第4章

步进电机的控制

步进电机(也称脉冲电机)是一种跟踪给定脉冲信号转动的电机。因此,它与上一章学习过的直流电机不同,即单纯向它施加电压是不会导致转动的。换句话说,要使它转动必须借助控制电路,在这个控制电路中,往往需要微控制器或专用芯片。

步进电机能根据给定的脉冲信号实现精确的定位控制,而且即使在停止时也有制动转矩,这些特性对机器人控制都是很有利的。

本章将学习步进电机的结构以及相关的驱动方法。

4.1 步进电机的基础知识

步进电机是一种能够根据脉冲(通常为方波)控制转角和转速、并适合微控器控制的电机。

4.1.1 步进电机的组成及原理

图 4.1 是二极的单极性步进电机的工作原理图。在永久磁铁转子的周围,按间隔 90°分布着 4 个定子。若如图 4.1(a)所示那样,定子 A 的电磁线图(a)被励磁,则转子的 N 极与定子 A 的 S 极相吸,电机静止。接着,若励磁从定子 A 切换到 B,那么如图(b)所示,转子的 N 极跟随定子 B 的励磁向前转动 90°。

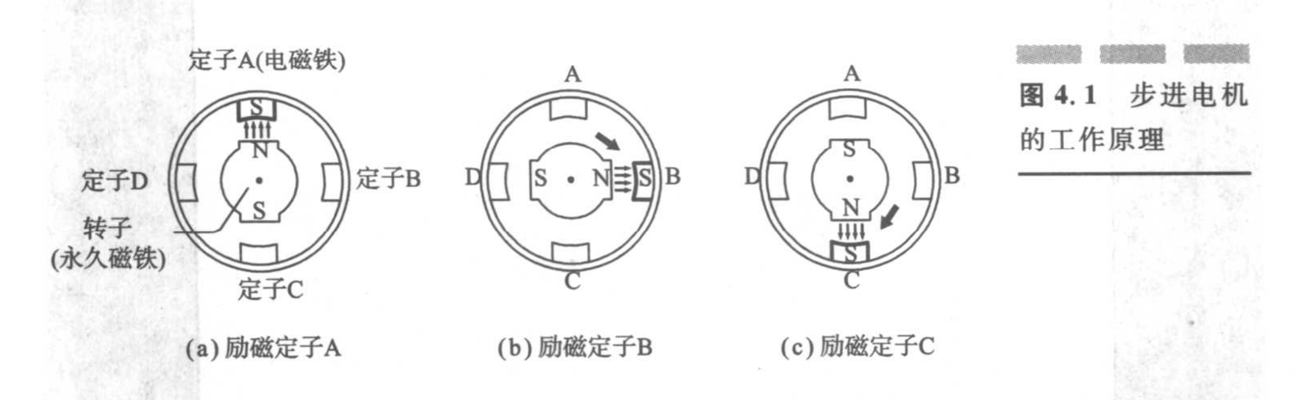


图 4.1　步进电机的工作原理

这样,如果定子按照 A→B→C→D→A 顺序切换励磁,则转子将沿着这个顺序以 90°的步长连续转动。也就是说,为此图 4.2 电路应该依照 S_1～S_4 的顺序切换开关的 ON 状态。

实际上,增加转子的极数,或者增加定子分布的密度都能减少单位步长所相当的转角。例如,若步进角是 1.8°每步的步进电机,1 个脉冲相当于 1.8°,50 个脉冲能转 90°。因此,步进电机可通过输入脉冲的个数来控制转角,再由脉冲周期控制转速,如图 4.3 所示。

图 4.2 步进电机的电路

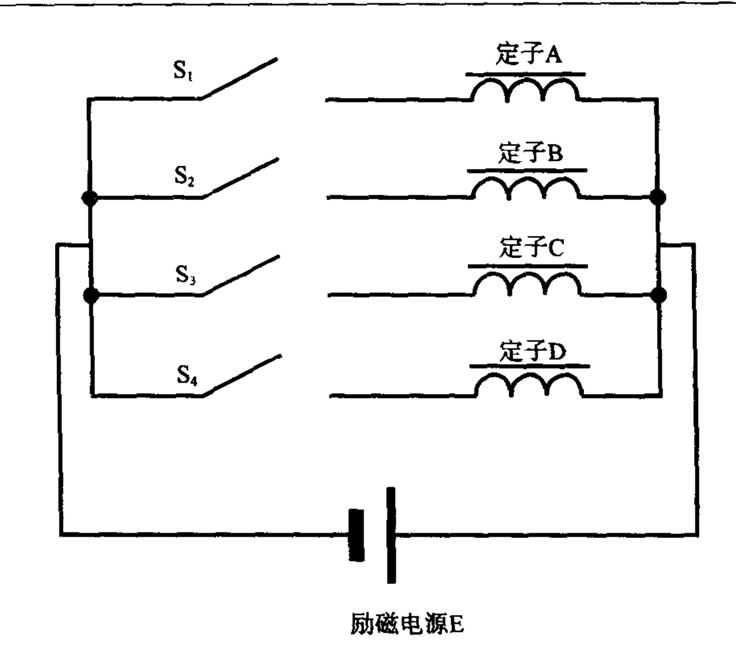
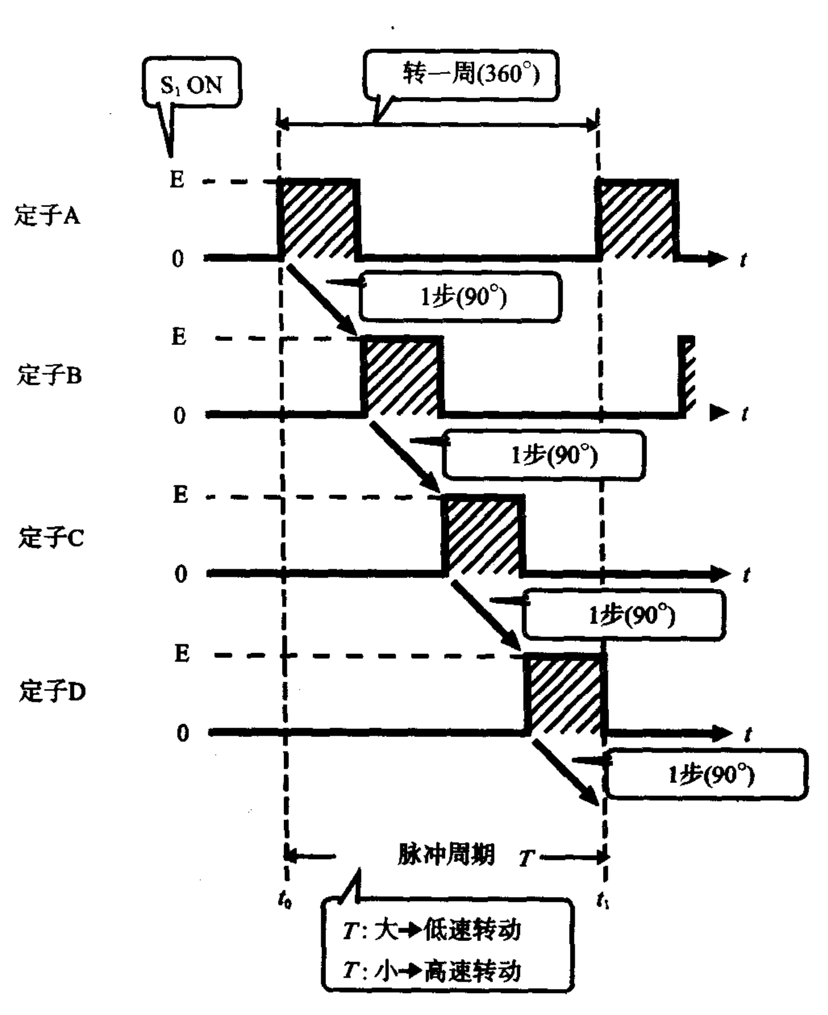


图 4.3 转角与转速



不过想让步进电机超过某个转速运转时，若从一开始起动就给出极短的脉冲周期，转子可能跟不上信号的输入。所以应该像图 4.4 那样在运行的起始阶段设

定加速过程，而在运行的结束段设定减速过程。

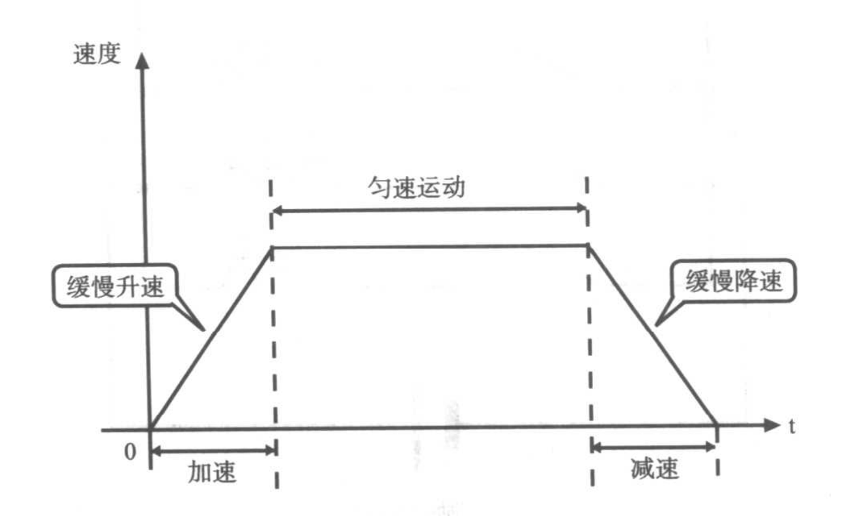


图 4.4　步进电机的速度控制

图 4.5 给出了各种步进电机的外形。可以把步进电机的特点归纳如下。

◆ **步进电机的特征：**

· 即使不用传感器，也能精确定位。
· 根据给定的脉冲周期，能够以任意速度转动。
· 即使处于停止状态，对定子励磁也能保持制动转矩。
· 难以获得较大的转矩。
· 不宜用于高速转动。

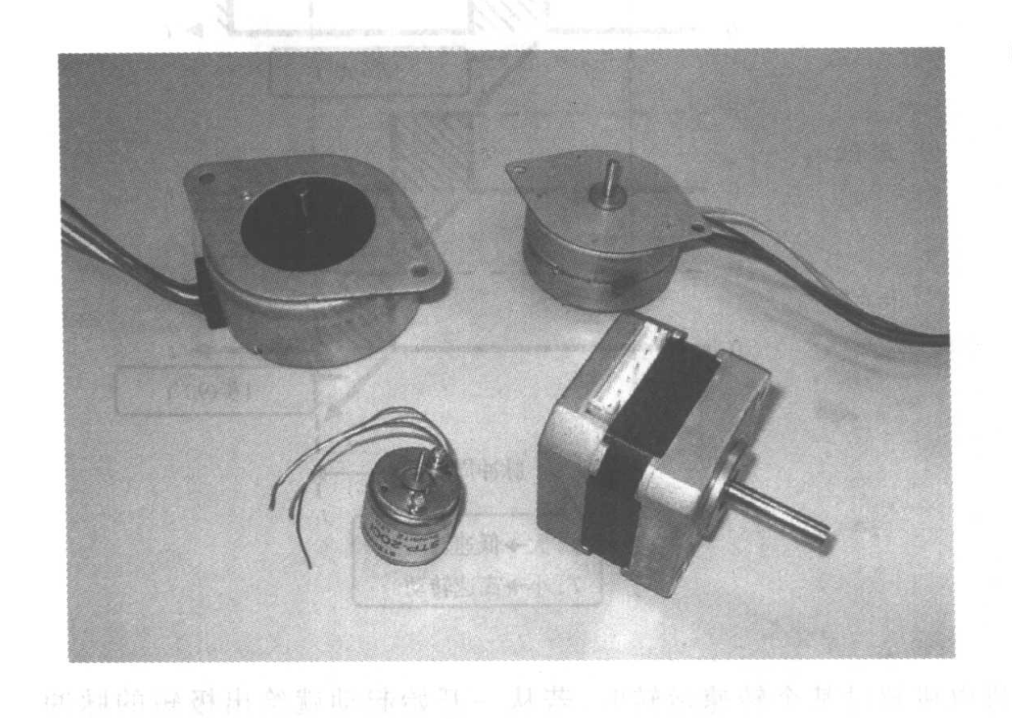
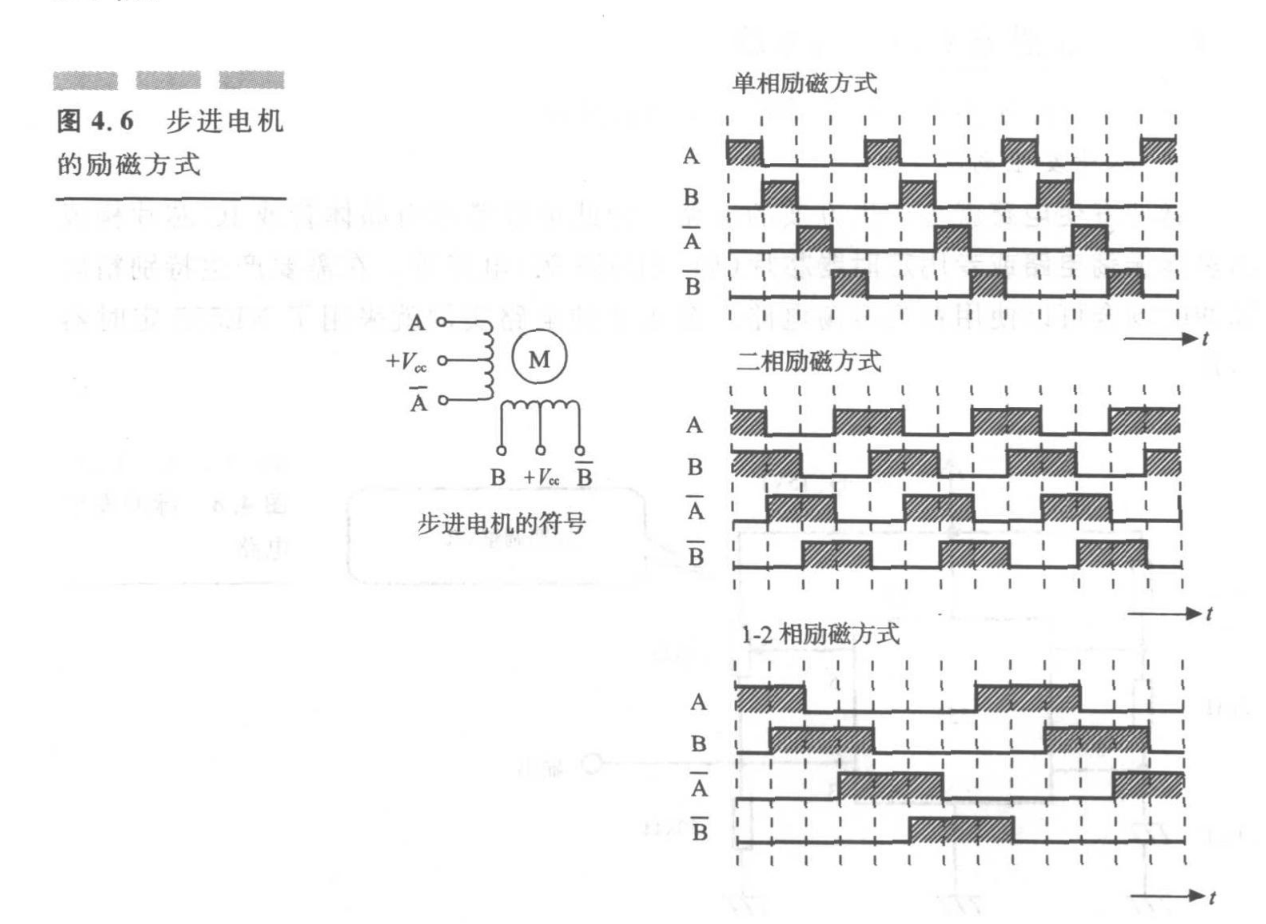


图 4.5　步进电机的外形

4.1.2 步进电机的励磁方式

图 4.3 所示的步进电机在某一时刻只有一个定子处于励磁状态，这种转动方式被称之为单相励磁方式。此外，还有二相励磁方式及 **1-2 相励磁方式**。图 4.6 表示步进电机各种励磁方式的脉冲时间顺序表。从图 4.6 可知，在二相励磁方式中相邻两个定子同时被励磁，因此，与单相励磁方式比较它的输出转矩为前者的 2 倍，功率消耗当然也是 2 倍。在 1-2 相励磁方式中，单相励磁和二相励磁相互交替进行，因此可以得到单相励磁方式的半个步距角，消耗功率则为单相励磁方式的 1.5 倍。

图 4.6 步进电机的励磁方式



4.2 步进电机的转动控制

使用微控制器能使步进电机的转动控制得到简化。本节就来学习控制步进电机转动的方法。

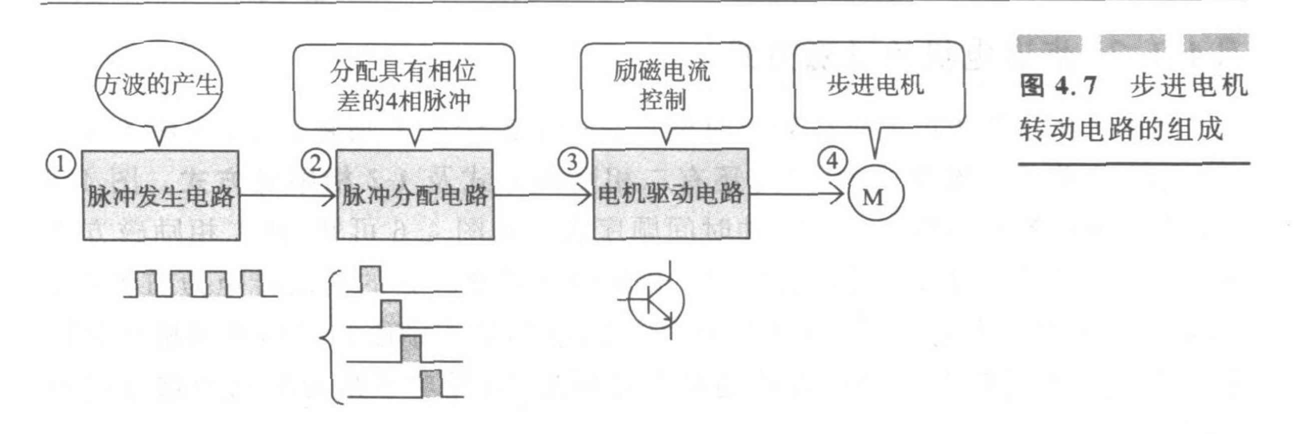


图 4.7　步进电机转动电路的组成

4.2.1　步进电机的控制电路

图 4.7 介绍了步进电机转动控制电路的组成。

1. 脉冲发生电路

脉冲发生电路就是产生方波的电路。为此可以考虑由晶体管或 IC 芯片构成的**多谐振荡电路**或专用**定时器芯片**(如 WE555 等)电路等。在需要产生特别精确脉冲的场合可以使用石英振荡电路。图 4.8 的电路实例就采用了 NE555 定时器芯片。

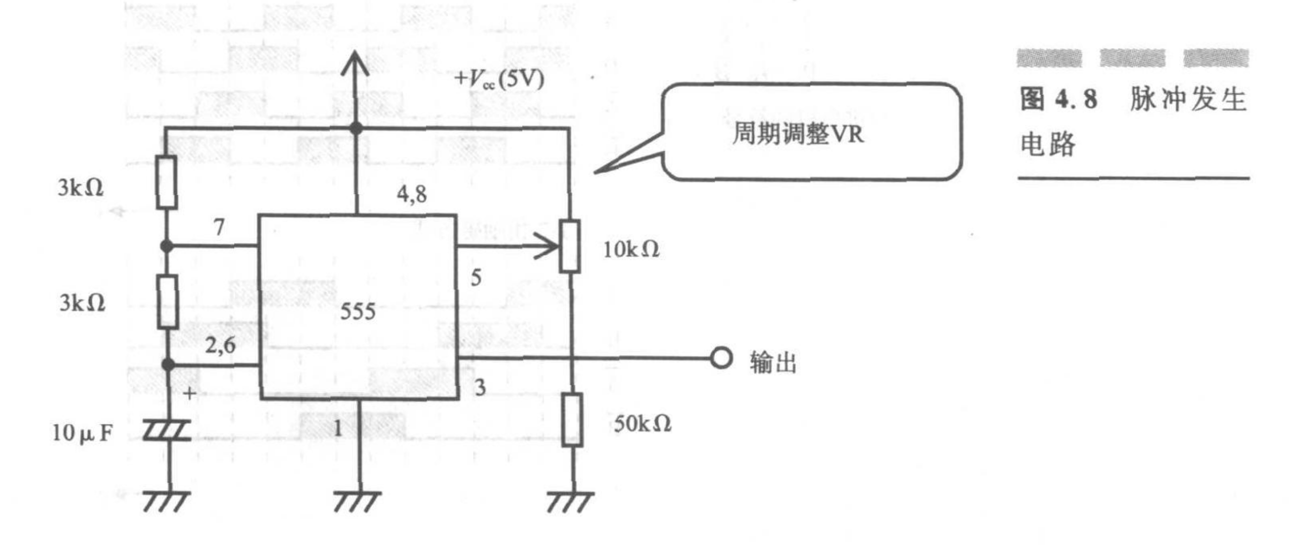


图 4.8　脉冲发生电路

2. 脉冲分配电路

这里以**步进电机**单相励磁方式转动为例。此时需要发生彼此相隔 90°相位的四相脉冲。**脉冲分配电路**的任务就是从脉冲发生电路产生的方波中分解出对应的四相脉冲。除采用**触发 IC 芯片**(如 74HC74 等)以外,市场上还有专用 IC 芯片(如 PMM8713 等)出售。图 4.9 是采用 **74HC74** 的二相励磁脉冲分配电路的例子。

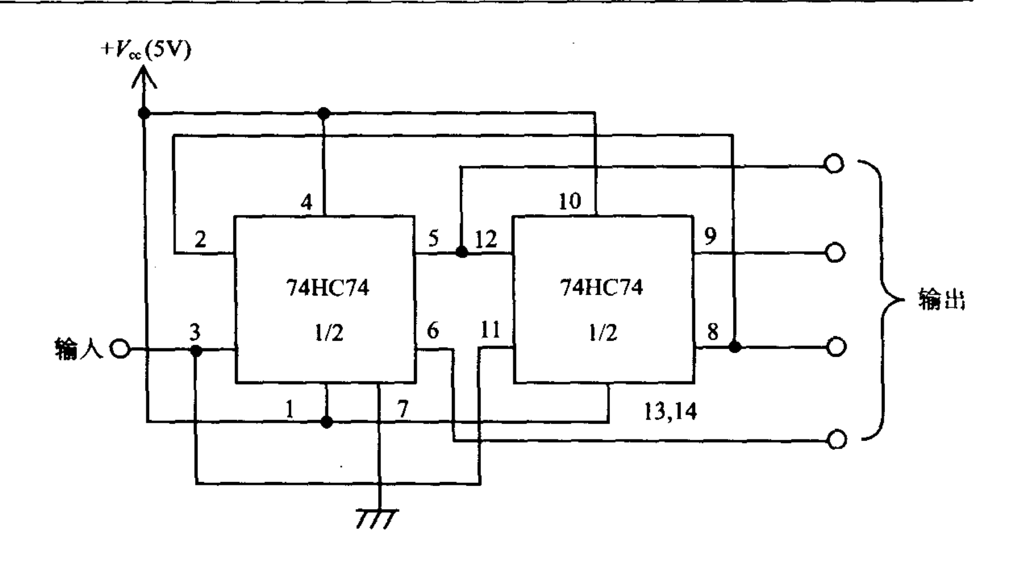


图 4.9 脉冲分配电路

3. 电机驱动电路

图 4.5 所示的步进电机中，各相约有几百毫安的励磁电流通过。因此，需要借助**功率三极管**或 IC 芯片驱动励磁电流。图 4.10 中是三极管电机驱动电路的实例。

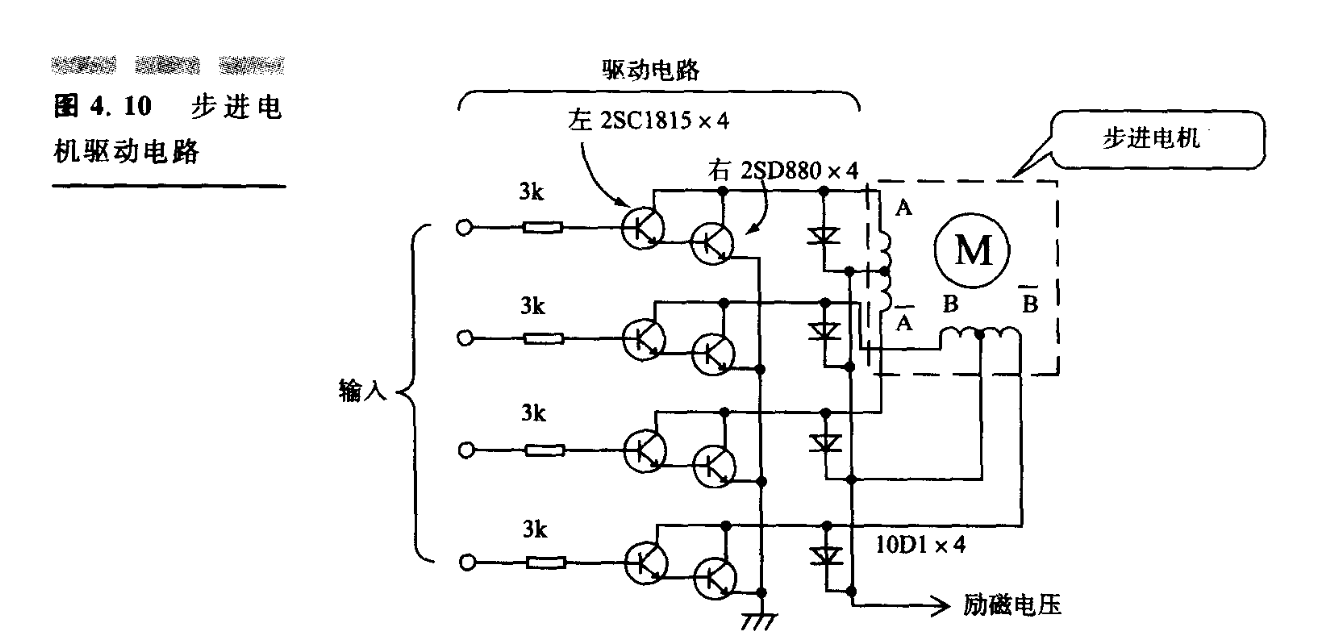


图 4.10 步进电机驱动电路

如图 4.10 所示，把两个晶体管组合起来的连接方法叫达林顿连接法，用在电流放大上面很有效。图 4.10 中的 2SD1415 实际上是已经将 2 个晶体管按达林顿连接方式封装起来的单元器件。图中与步进电机连接的二极管的作用是用来吸收定子线圈产生的反电动势。

4. 步进电机

应该根据不同的用途来选择合适的步进电机，如考虑步距角、励磁电流、转矩、

外形等因素。

图 4.8、图 4.9 的电路虽然可以使步进电机转动，但是仍无法实现精确定位和速度控制的目的。为此，下面学习用微控制器进行电机定位和速度控制的方法。

4.2.2 基于微控制器的控制

在图 4.7 所示的步进电机转动电路中，用微控制器取代图 4.8 脉冲发生电路和图 4.9 脉冲分配电路的部分功能，就成为图 4.11 所示的微控制器步进电机控制系统。

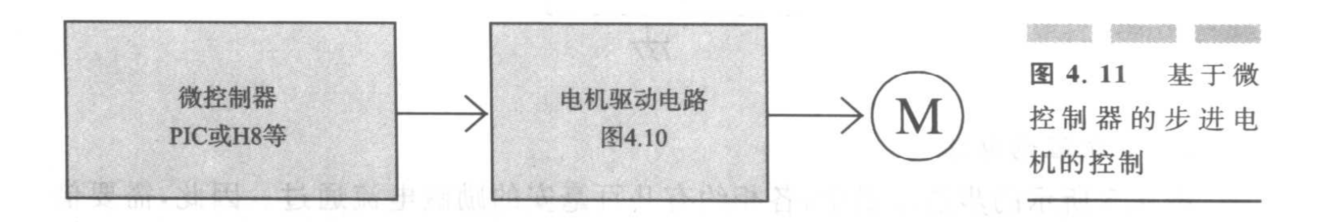


图 4.11　基于微控制器的步进电机的控制

根据给定的微控制器程序，适当地改变电机驱动电路的信号，可以实现相应的定位控制和速度控制。

下面以 PIC 微控制器为例开展步进电机转动控制的实验。

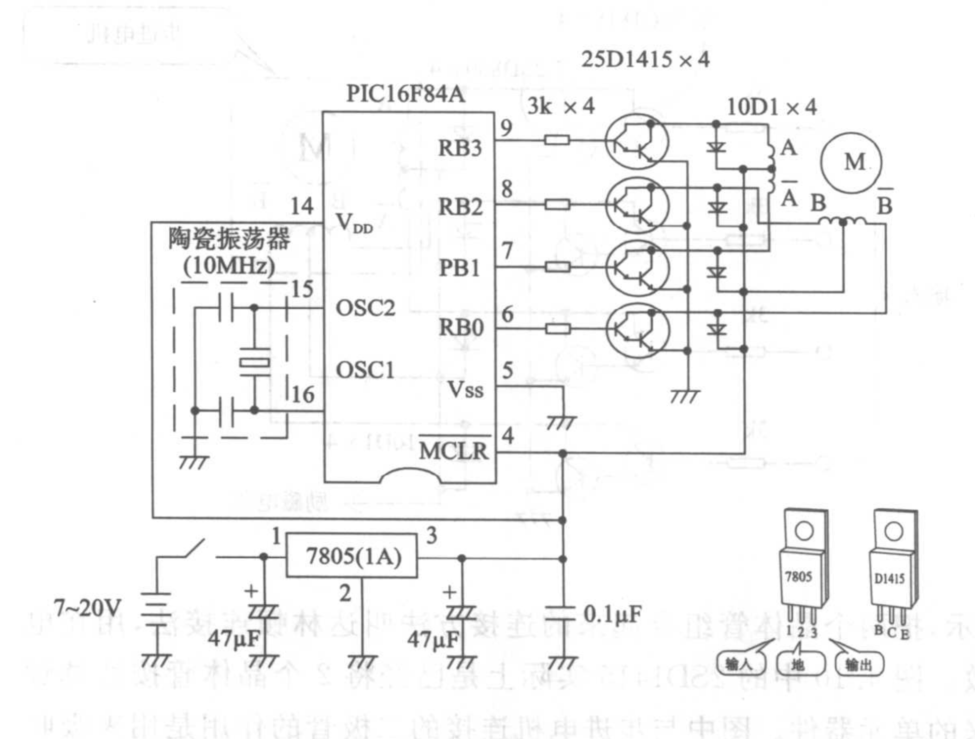


图 4.12　基于 PIC 的步进电机控制电路

1. 电路组成

图 4.12 的电路基于第 2 章介绍的 PIC16F84A 对步进电机进行控制。

设置 PIC 的端口 B 的 4 位(RB0～RB3)为输出，将之与**电机驱动电路**连接。根据该

4位端口输出信号的内容控制步进电机。本例步进电机的步距角为7.5°,每相的励磁电流为220mA。由于励磁电源的电压为5V,因此可以与微控制器的电源共用。

图4.13是实际电路与步进电机的外形。

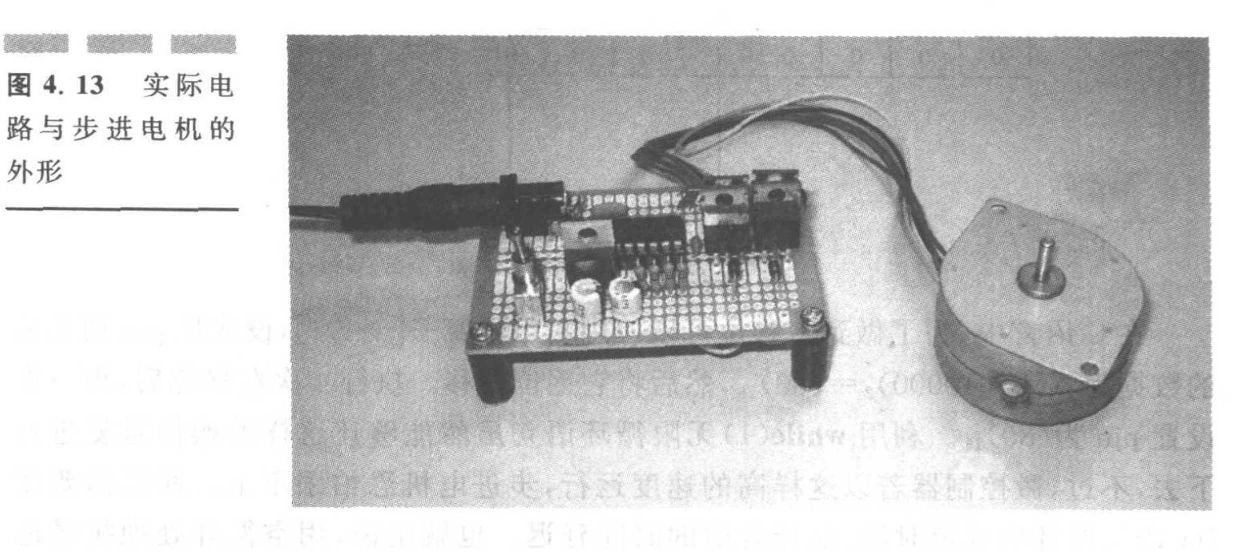


图4.13 实际电路与步进电机的外形

2. 单相励磁方式

单相励磁方式是每次只对某一个定子励磁,彼此不断地连续轮流励磁。显然,为此应该向图4.14中PIC微控制器的输出端口B的各位(RB0～RB3)轮换地输出数据"1"。

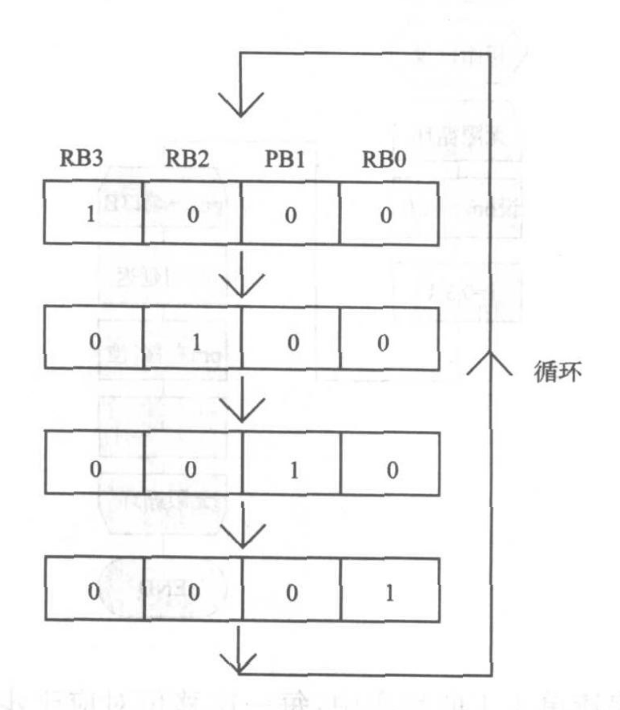


图4.14 单相励磁输出模式

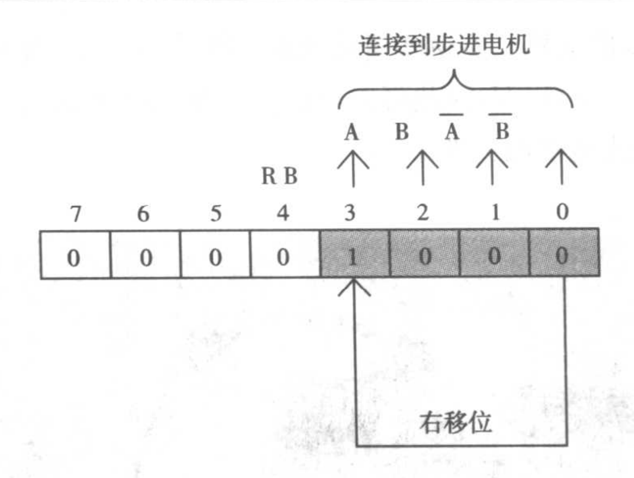


图 4.15　由右移位进行单相励磁

在 C 语言中，为了做到右移位可以使用**右移位算子**[> >]，设变量 pm 初始值的数据为：$(000010000)_2=(80)_{16}$，然后将它逐位右移。执行 4 次右移位后，再一次设置 pm 为$(80)_{16}$。利用 **while**(1)无限循环语句虽然能够让这样的操作重复进行下去，不过，微控制器若以这样高的速度运行，步进电机恐怕跟不上。所以需要用 **for** 语句循环构成定时器，执行合适的时间延迟。也就是说，用空循环处理缓慢地进行移位，以便满足步进电机的响应。

图 4.16 是**单相励磁**的流程图。程序清单 4.1 是基于 PICC Lite 的源程序。

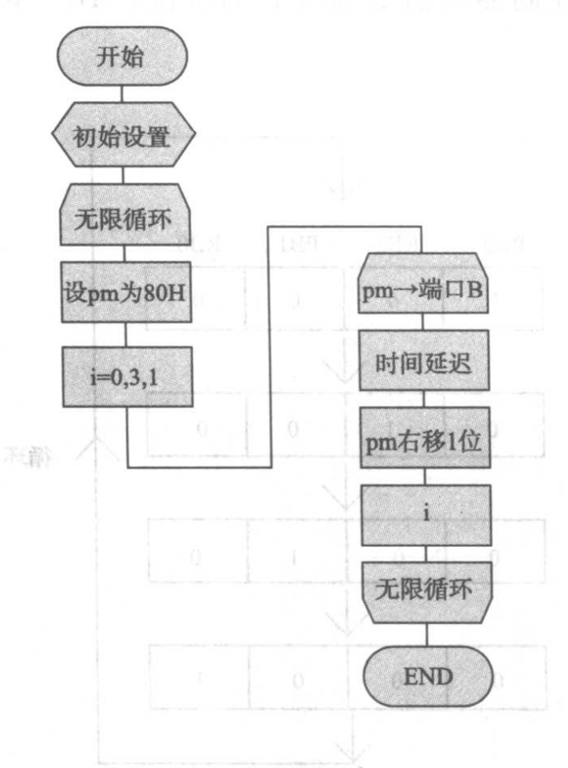


图 4.16　单相励磁的流程图

在程序清单 4.1 的程序中，每一次移位对应于步进电机转动一个步距角。因此，若想进行步进电机的定位控制，只须计算移位执行的次数即可。另外，若要进

行速度控制，只须增加或减少 for 语句中计数器变量的值。计数器变量的值越小，步进电机的转动速度就越高。在本例的这个程序中，由于步进电机的转速设定得比较低，所以，可以暂不考虑本书曾经在图 4.4 中说明的缓慢上升、缓慢下降的运动加减速要求。

当希望电机的转动方向相反，应该把端口 B 的初值设定为 $(00010001)_2 = (11)_{16}$，并按照左移位[< <]运算，电机即可向左转动。

```
// 程序清单 4.1
// 步进电机控制程序(单相励磁)

#include <pic.h>                          // 头文件

main()
{
      unsigned int pm;                    // 定义无符号 8 位整数 pm
      unt i;                              // 定义计数器变量
      long c;
      TRISB = 0x00                        // 设定端口 B 为输出

      while(1){                           // 重复循环
            pm = 0x08;                    // 设定转动数据
            for(i = 0;i<4;i + +){         // 循环移位
              PORTB = pm;                 // pm→输出到端口 B
              for(c = 0;c<30000;c + +)    // 时间延迟
                ;
              pm = pm> >1;                // 右移 1 位
            }
      }
}
```

程序清单 4.1　单相励磁程序(PICC Lite)

3. *二相励磁方式*

下面，改用二相励磁方式使步进电机转动。使用的电路与上面单相励磁方式相同(参见图 4.12)。在二相励磁中，定子按照 $AB \to B\overline{A} \to \overline{AB} \to A\overline{B} \to AB$ 的顺序励磁。图 4.17 是根据右移位进行二相励磁的情形。

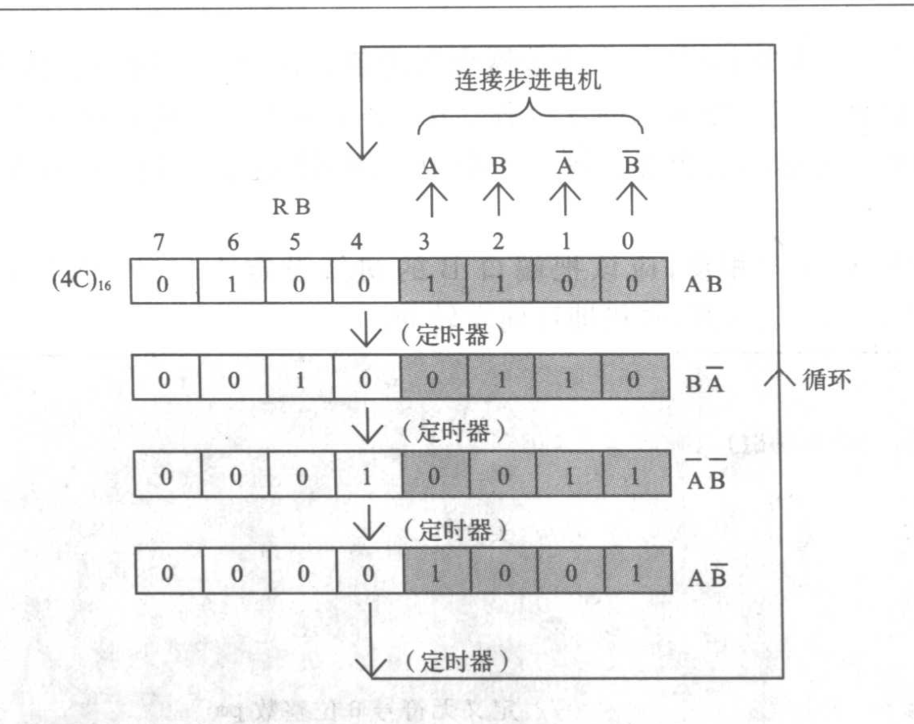


图 4.17　根据右移位的二相励磁

使用右移位算子[＞＞]，设定 pm 的初始值为$(01001100)_2=(4C)_{16}$，逐位向右移动。经向右 4 次移位之后，再一次把 pm 初始化为 pm$=(4C)_{16}$，接着继续向右移位，程序清单 4.2 给出了它的源程序。可以看出，程序清单 4.1 与程序清单 4.2 之间的不同仅仅在于改变了变量 pm 的初始值，结果也就改变了步进电机的励磁方式。

实际运行中，若在步进电机转动状态下尝试握一握转子的轴，应该可以感觉到比单相励磁更大的转矩。

```
// 程序清单 4.2
// 步进电机控制程序(二相励磁)

#include<pic.h>                                // 头文件

main()
{
        unsigned int pm;                       // 定义无符号 8 位整数 pm
        int i;                                 // 定义计数器变量
        long c;
        TRISB = 0x00;                          // 设定端口 B 为输出

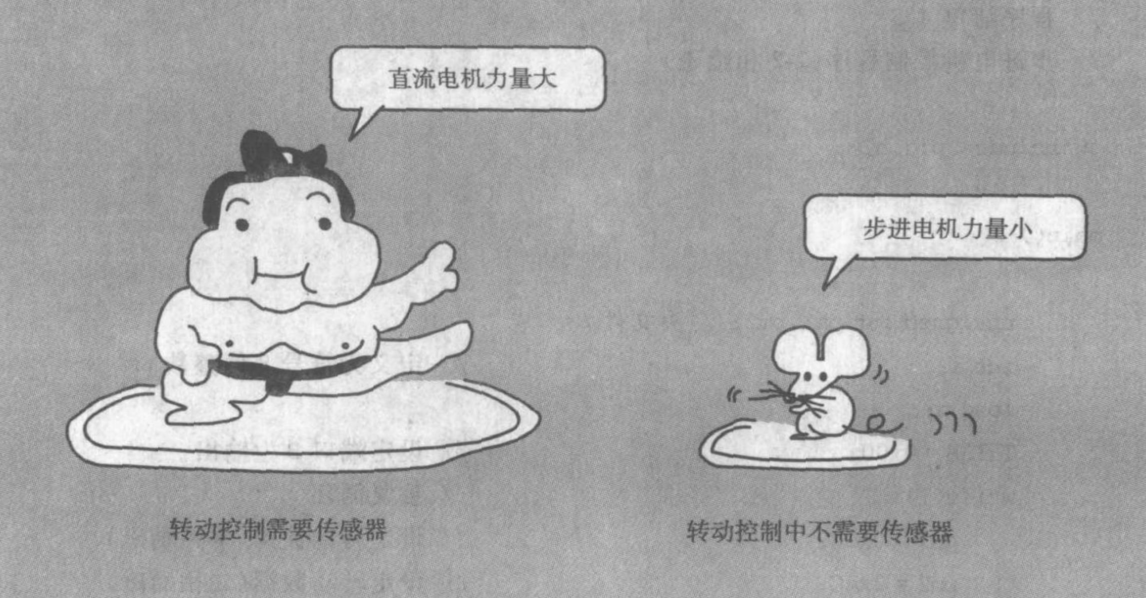
        while(1){                              // 重复循环
              pm = 0x4c;                       // 设定转动数据
```

```
        for(i=0;i<4;i++){                         // 循环移位
                PORTB=pm;                          // pm→输出到端口 B
                for(c=0;c<30000;c++)               // 时间延迟
                        ;
                pm=pm>>1;                          // 右移 1 位
                }
        }
}
```

程序清单 4.2 二相励磁程序(PICC. Lite)

专栏

自主机器人相扑比赛是由搭载微控制器的两台机器人相扑选手在摔跤台上进行的角力竞技,日本从 1990 年开始举行全国性比赛。比赛的初期,曾经看到过步进电机驱动机器人的例子,但是目前,都改成直流电机驱动。也就是说,步进电机不适宜应用在相互推挤的相扑机器人上。但是另一方面,在精确定位控制十分关键的微型鼠的比赛中,使用步进电机驱动机器人的事例却屡见不鲜,甚至是主流。另外,在打印机、硬盘等诸多办公设备上也经常采用步进电机。



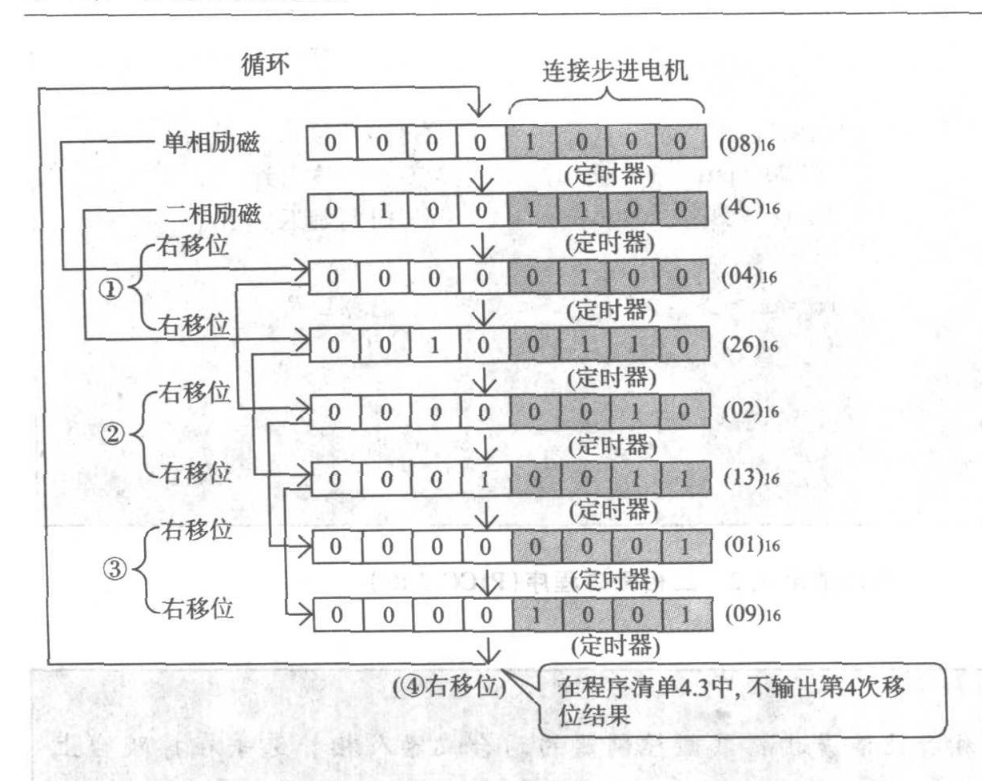


图 4.18　采用右移位的 1-2 相励磁方式

4. 1-2 相励磁方式

1-2 相励磁方式如图 4.18 所示，它通过单相励磁与二相励磁相互交替重复使步进电机转动。程序清单 4.3 中给出了基于 PICC Lite 的源程序。

```
// 程序清单 4.3
// 步进电机控制程序(1-2 相励磁)

#include<pic.h>

main()
{
        unsigned int pm1,pm2;// 头文件
        int i;                                   // 定义无符号 8 位整数 pm
        long c;
        TRISB = 0x00;                            // 设定端口 B 为输出
        while(1){                                // 重复循环
            pm1 = 0x08;                          // 设定转动数据(单相励磁)
            pm2 = 0x4C;                          // 设定转动数据(二相励磁)
            for(i = 0;i<4;i + +){                // 循环移位
                PORTB = pm1                      // pm1→输出到端口 B
```

```
                    for(c = 0;c<30000;c + +)              // 时间延迟
                          ;
                    PORTB = pm2;                          // pm2→输出到端口 B

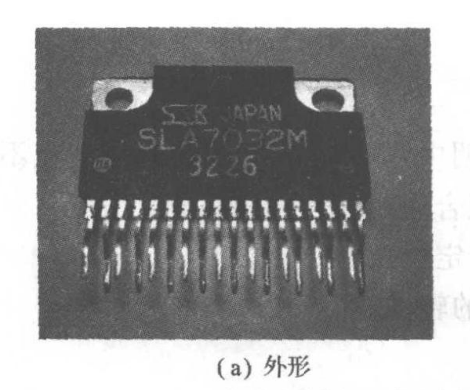
                    for(c = 0;c<30000;c + +)              // 时间延迟
                          ;
                    pm1 = pm1>>1;                         // 右移 1 位(单相数据)
                    pm2 = pm2>>1;                         // 右移 1 位(二相数据)
          }
     }
}
```

程序清单 4.3　1-2 相励磁程序(PICC. Lite)

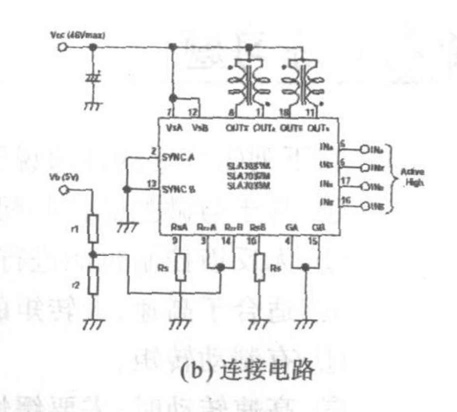
4.2.3 基于专用芯片(IC)的控制

现在,市面上有各种步进电机控制专用芯片出售。这里,以 3K 电子株式会社生产的**二相步进电机驱动芯片 SLA7032M** 为例介绍如何使用专用芯片控制步进电机的方法。图 4.19(a)为 IC 的外形,图 4.19(b)为连接电路举例。

图 4.19 SLA7032M (3K 电子株式会社提供)



(a) 外形



(b) 连接电路

这个芯片内藏的驱动电路(参见图 4.10)可控制最大 46V、1.5A 的步进电机。励磁模式是向 4 条 IN 引脚输入。表 4.1 为二相励磁方式,表 4.2 为 1-2 相励磁方式驱动电机转动时的输入模式。

clock	0	1	2	3	0	1
IN_A	H	L	L	H	H	L
$IN_{\bar{A}}$	L	H	H	L	L	H
IN_B	H	H	L	L	H	H
$IN_{\bar{B}}$	L	L	H	H	L	L

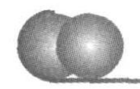
表 4.1　二相励磁方式的输入

clock	0	1	2	3	4	5	6	7	0	1	2	3
IN_A	H	H	L	L	L	L	L	H	H	H	L	L
$IN_{\bar{A}}$	L	L	L	H	H	H	L	L	L	L	L	H
IN_B	L	H	H	H	L	L	L	L	L	H	H	H
$IN_{\bar{B}}$	L	L	L	L	L	H	H	H	L	L	L	L

表 4.2　1-2 相励磁方式的输入

另外，改变施加在输入引脚 $R_{EF}A$ 和 $R_{EF}B$ 的电压，可以改变步进电机的电流。根据这个原理，步进电机的转矩和功率损耗就能够被调节。

采用上面列举的专用芯片，可以减少元器件的数量，同时提高电路的可靠性。



练习题

1. 在下列①～⑤的各项说明中，哪一项与步进电机的特点不相符？

   ① 易于与微控器控制配合应用。

   ② 无反馈控制也能进行定位控制。

   ③ 适合于高速、大转矩的转动。

   ④ 有制动转矩。

   ⑤ 高速转动时，需要缓慢加速和缓慢减速。

2. 有一个步距角为 7.5°的步进电机，让其转动 90°，需要步进电机工作几步？

3. 在图 4.20 所示时间序列表中，如果把相应的数据输出到步进电机，会产生何种转动的方式？

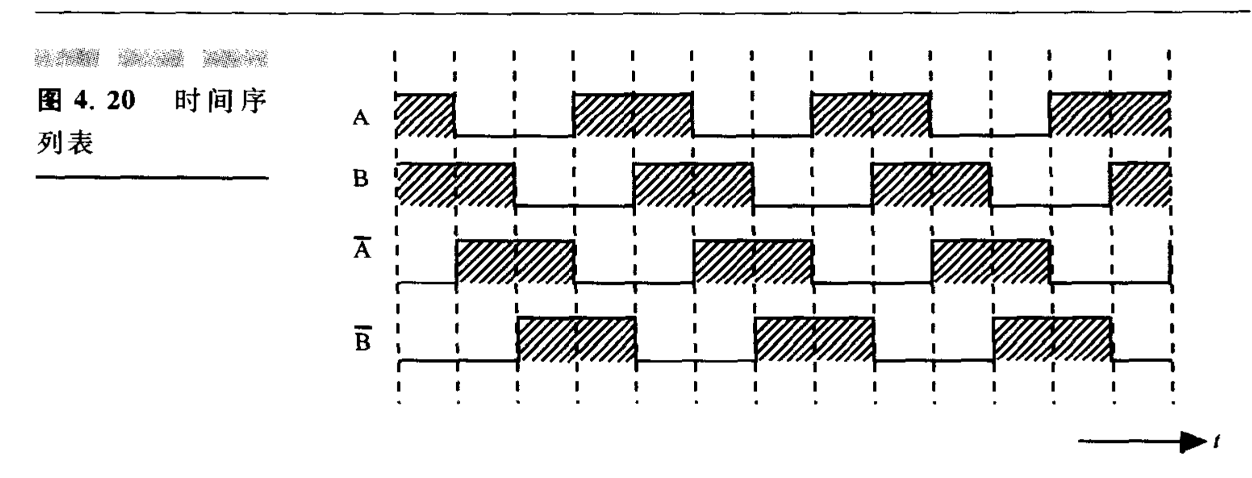
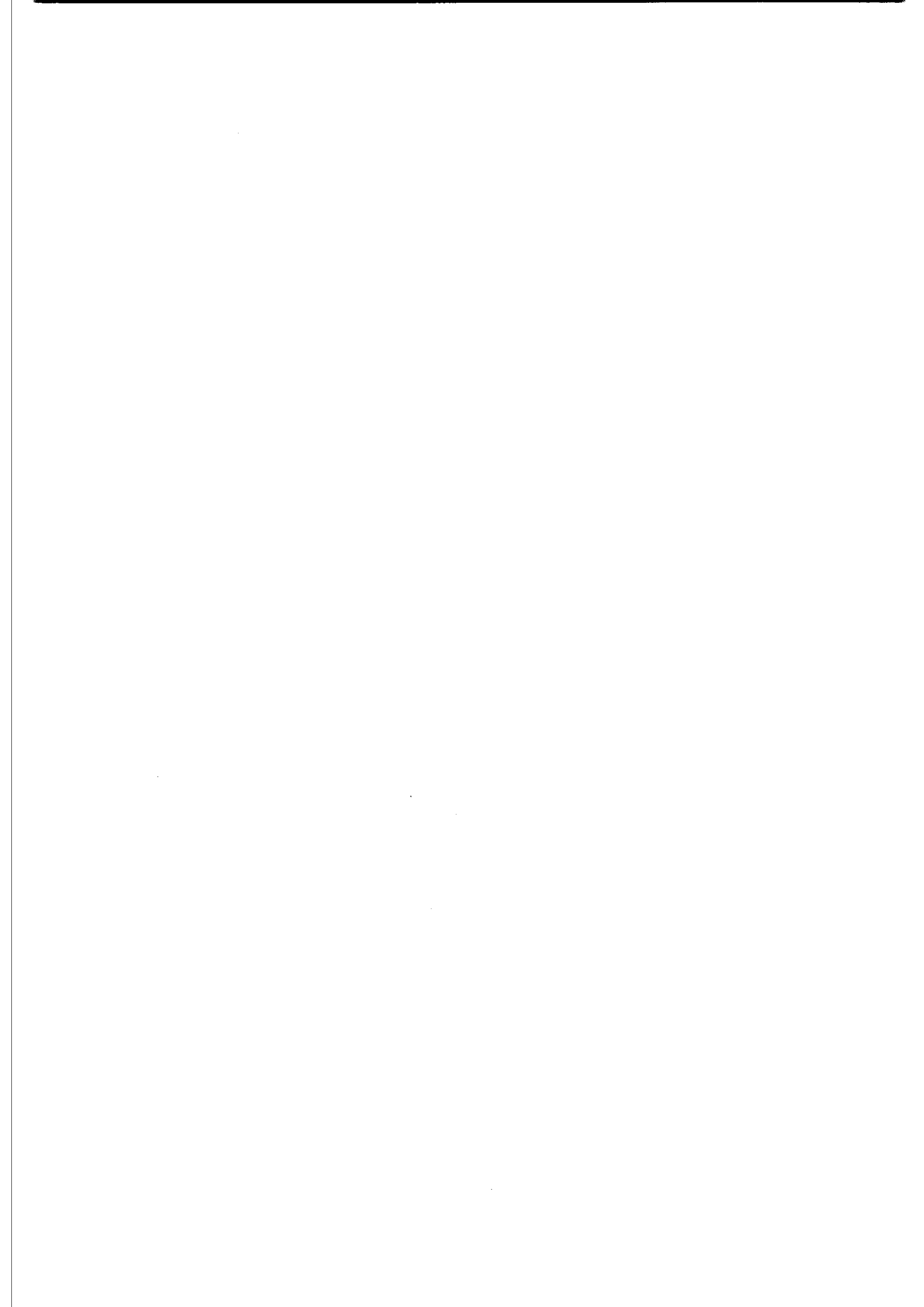


图 4.20 时间序列表

4. 简述各种励磁方式(单相、二相、1-2 相)的特点。
5. 用微控制器控制步进电机转动时,说明借助计时器实现时间延迟的原理。
6. 简述在 C 语言中,位算子的移位操作是怎样实现的。
7. 简单说明选用步进电机专用驱动芯片的优点。
8. 在步进电机专用芯片中,有调整步进电机电流的功能,试举例说明这个功能的应用。



第5章

传感器

人类通过五官，即看、听、嗅、味、触来获取外界信息。

但是计算机、电子电路却只能处理电子信号，不能像人类那样直接从外界获取信息，而传感器却具有类似人类五官感觉的功能，它们是将外界信息转换成电子信号的电子元器件。

其实，传感器所能处理的信息不仅限于相当人类五官感觉的外部信息，而且还能处理温度、压力、流量等物理量。机器人必须一边对外界信息作出判断，一边采取有效的处理措施，从这个意义上说，传感器是机器人不可缺少的电子元器件之一。

本章将学习传感器的工作原理及使用方法等基础知识。

5.1　传感器的基础

提到传感器，除了机器人，在我们身边的电气产品、摄像机、打印机，自动售货机、报警器等等诸多产业中它的用量不计其数。例如，图 5.1 所示的双足步行机器人就用到了表 5.1 所示的角加速度传感器、加速度传感器、距离传感器、足底传感器、**触觉传感器**等多种传感器。

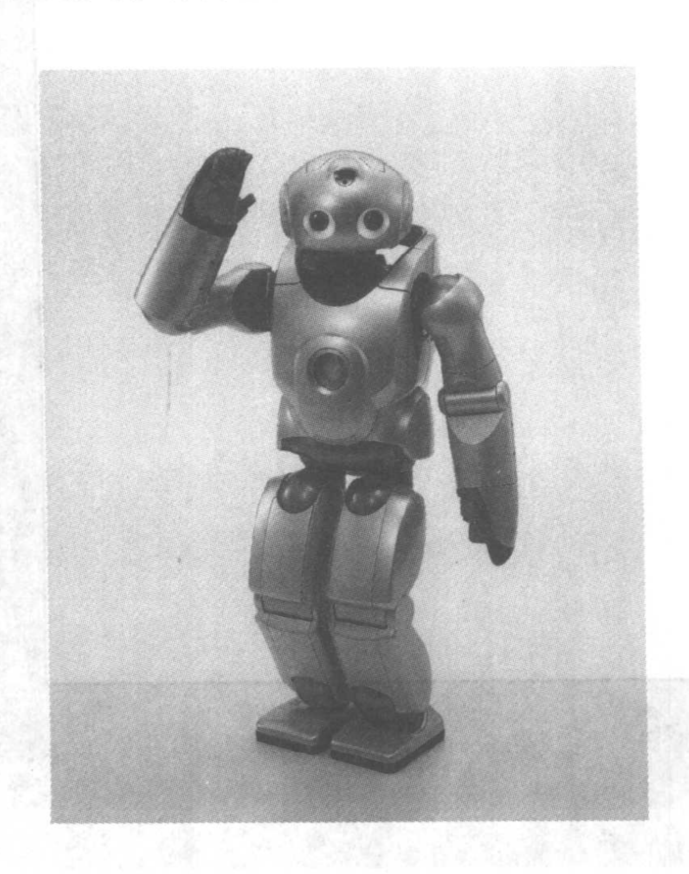


图 5.1　双足步行机器人 QRIO 照片（摘自索尼株式会社）

表 5.3 双足步行机器人传感器应用举例（摘自 QRIO）

测距传感器	红外线方式：头部、手部
加速度传感器	躯干：X，Y，Z/3 轴、足部：X，Y/2 轴
角速度传感器	躯干：X，Y，Z/3 轴回转
足底传感器	足部
温度传感器	手部、足部、头部、躯体、执行器
触觉传感器	头部、手部、肩部
内部传感器	全身各部
微动开关	背部

至于读者面对的自制机器人对象，其实传感器如此之多、控制如此复杂的例子并不多见，但这仍足以让我们充分了解到传感器对于机器人的重要性。

我们身边的数字摄像机和普通摄像机（参见图 5.2）也是传感器应用的典型例子。它们有的采用**红外线传感器**，通过对象反射的红外线检测距离，或者将 CCD 传感器的反差转换成电信号进行波形处理、聚焦。还有的摄像机内藏三维陀螺仪，或三维加速度传感器用来克服拍摄时手部的抖动。




图 5.2 数字摄像机和普通摄像机

再如打印机，纸张规格、纸屑、墨汁残量等等检测都靠传感器。表 5.2 列出了人类五官感觉与各种传感器之间的对应关系，五官感觉以外的传感器如表 5.3 所示。

表 5.2 人的五官感觉与各种传感器的对应关系

感　觉	物理/化学量	传感器用例
视觉（眼）	光	光电二极管、光电三极管、Cds、CCD 图像传感器、CMOS 图像传感器
听觉（耳）	音	麦克风
嗅觉（鼻）	气体	气体传感器、气味传感器
味觉（舌）	水质	pH 传感器
触觉（皮肤）	温度	热敏电阻、热电偶、半导体温度传感器
	压力	触觉传感器、半导体压力传感器、应变仪、感压传感器

表 5.3　五官感觉以外的传感器

检测量	传感器举例
红外线	热电红外传感器、热敏红外传感器 量子红外传感器(光电二极管、光电三极管)
超声波	超声波传感器
磁性	磁性传感器、霍尔元件、磁阻元件
速度、加速度	加速度传感器、G 传感、陀螺仪

可见传感器是如此多种多样，在此无法一一详述。本节仅对几种机械装置及机器人常用的具有代表性的传感器的工作原理加以说明，其中仅对最常用的光传感器及超声波传感器的具体使用方法进行了较为详细的说明。

5.1.1　触觉传感器

触觉传感器利用与物体的直接接触进行检测。具体地说，如图 5.3 所示的**微动开关**、**簧片开关**等，还有图 5.4 的传感器，它有一个触头，能检测来自其它方向的接触状态。不过它的构造原理与一般的按钮开关无异。

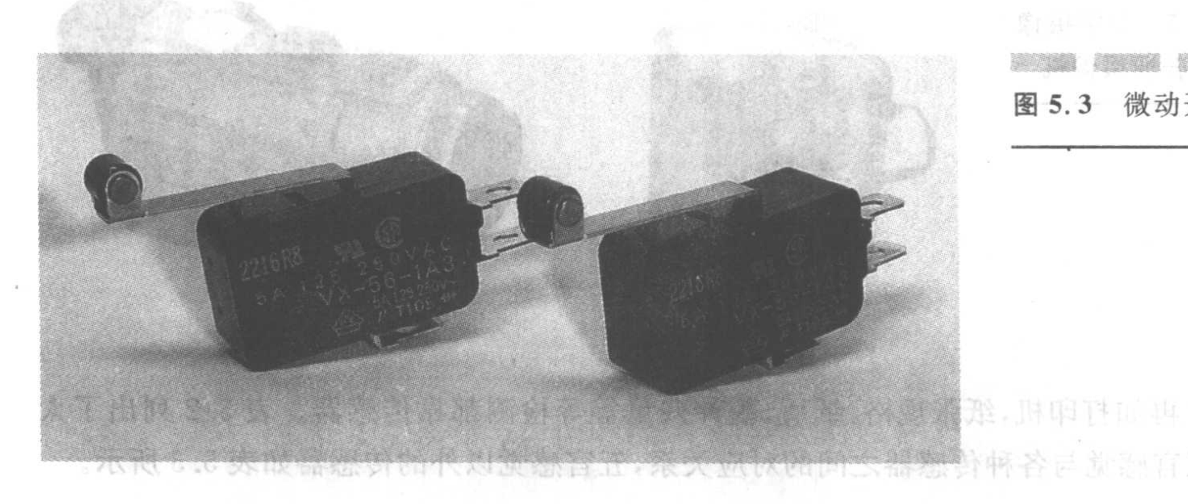


图 5.3　微动开关

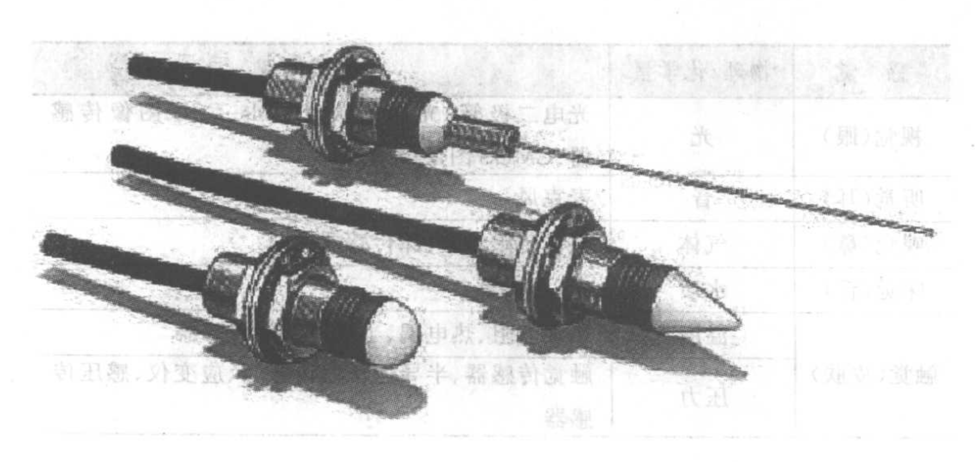


图 5.4　触觉传感器(摘自欧姆龙株式会社)

触觉传感器的电路(参见图 5.5)基本上类似于按钮开关电路。同样,由于触点的抖动现象会造成接通的时断时续,所以在使用这个电路时要注意克服。

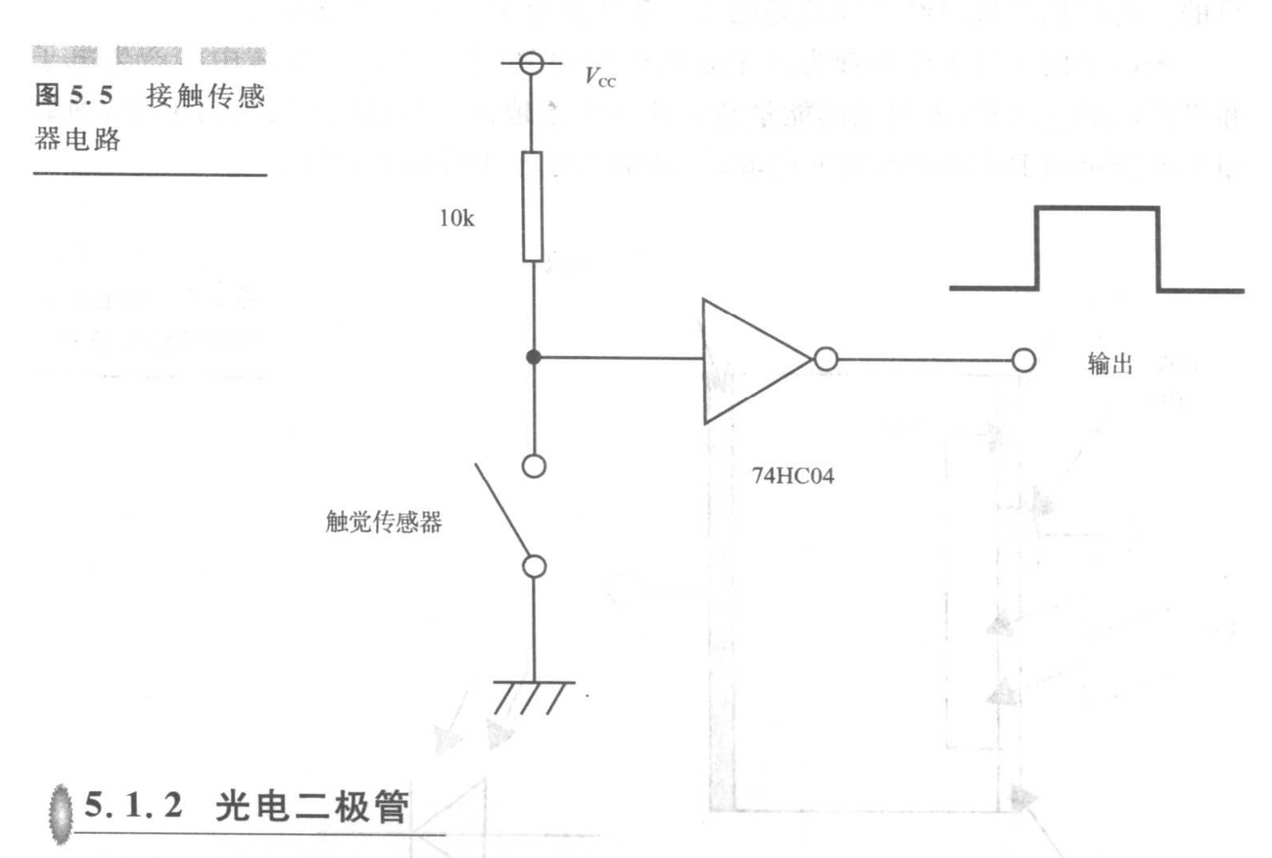


图 5.5 接触传感器电路

5.1.2 光电二极管

光电二极管(PD)是最典型的半导体光传感器,当光线照射到半导体的 PN 结时它的感光元件会产生电流或电压。光电二极管通常被用做检测光线强弱的变化,实物照片如图 5.6 所示。

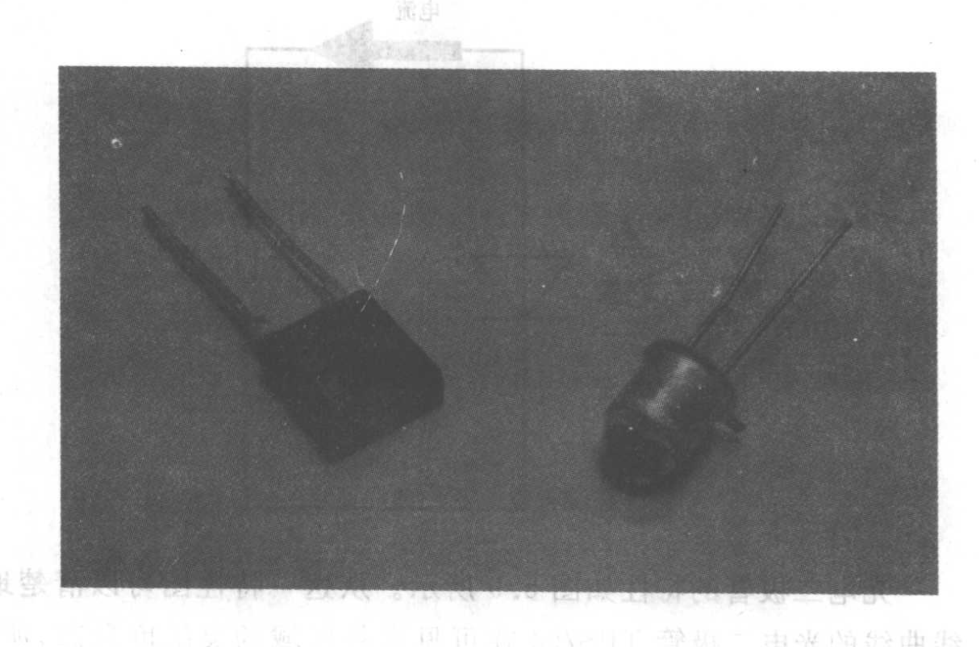


图 5.6 光电二极管照片(株式会社东芝:左为 TPS703,右为 TPS708)

光电二极管不但能检测可见光，还能检测红外线，有些特殊元件还能检测出紫外线。光电二极管的原理与太阳能电池基本相同，但太阳能电池是以提取能源为目的。有时也将检测红外线的光电二极管特别称为红外线传感器。

光电二极管的工作原理为当光线照射到硅 PN 结上时，Si 晶体中将产生电子和带正电的空穴对，入射光的能量越大这种现象越强，电流从 N 层流向 P 层，即如图 5.8 所示的 PN 结的相反方向流动，其结果是光能转换成电流。

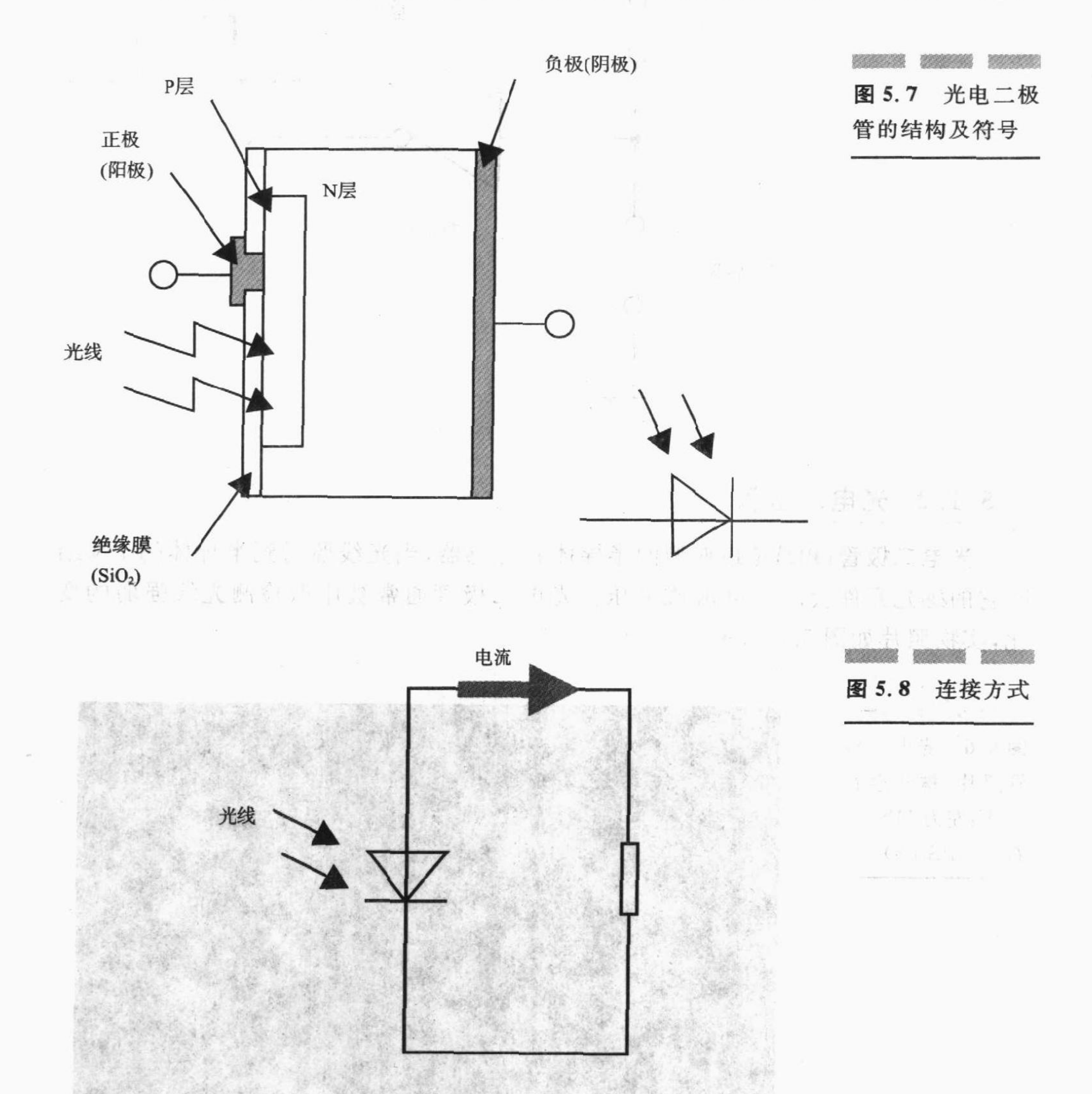


图 5.7 光电二极管的结构及符号

图 5.8 连接方式

光电二极管的特性如图 5.9 所示。从这一特性图可以清楚地看到，对应于实线曲线的光电二极管 TPS704 在可见光外区域的灵敏度最高，所以适合于充当紫

外线传感器。

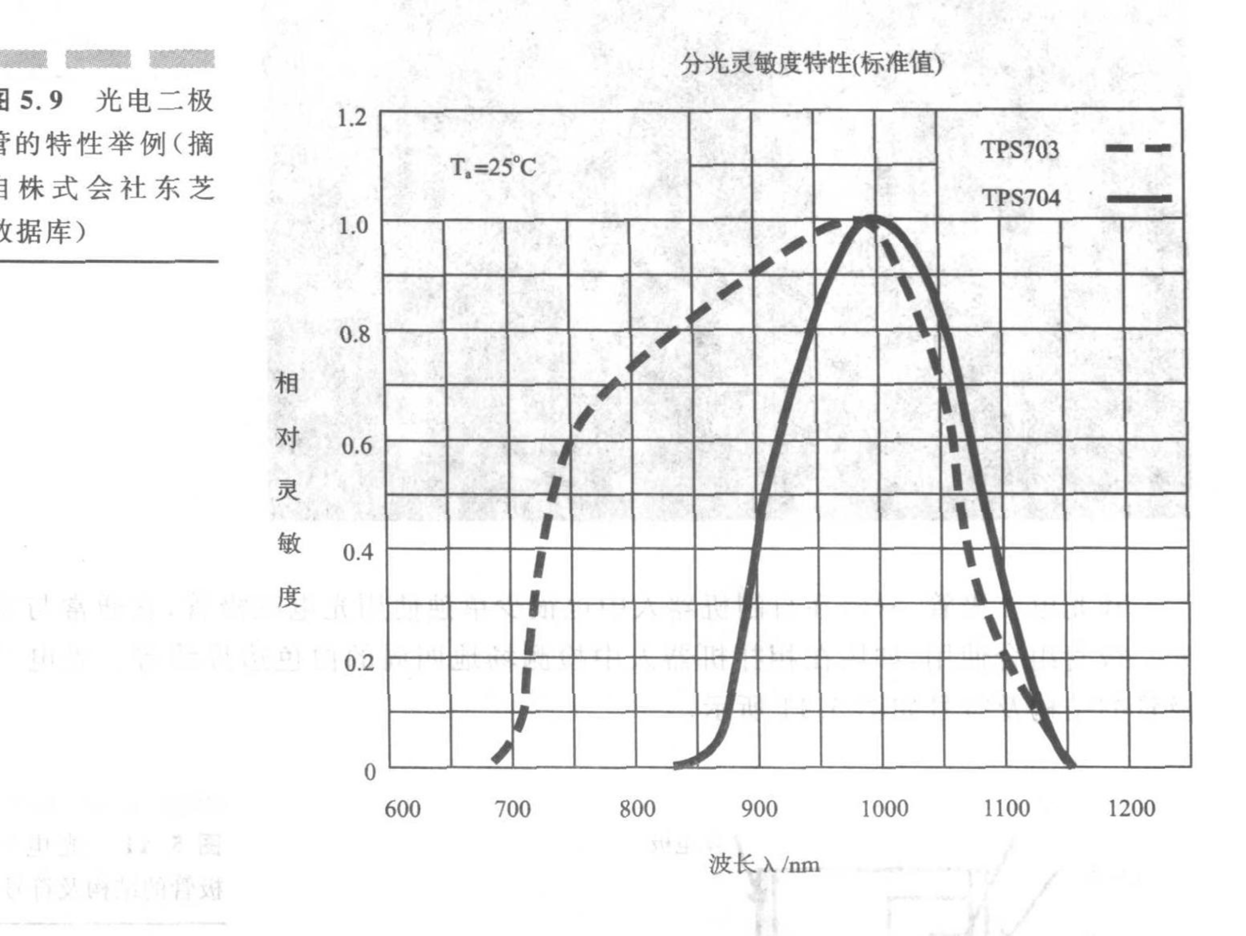


图 5.9 光电二极管的特性举例(摘自株式会社东芝数据库)

专栏:何谓可见光

可见光就是太阳光、日光灯光等人的眼睛能够感受到明亮的光线。人眼能够感觉到的光波被称为可见光,以便与红外线、紫外线等人眼无法辨认的光区别开来。可见光具有个人差别,一般指波长在 380～780nm 之间。1nm(纳米)=1×10^{-9}m。光线的波长具有如下关系:紫外线(短)<可见光<红外线(长)。

红外线遥控器的原理也是基于光电二极管。此外应用面很广的还有一体化的反射式**光电判读器**、**光电耦合器**等等。但是单个光电二极管的灵敏度很差,所以在自制机器人中很少采用。

5.1.3 光电三极管

光电三极管也只有两只管脚,外观上与光电二极管并无二致,它是在光电二极管的基础上再集成放大功能并封装在一块半导体芯片上制成的。图 5.10 为光电三极管的实物照片。

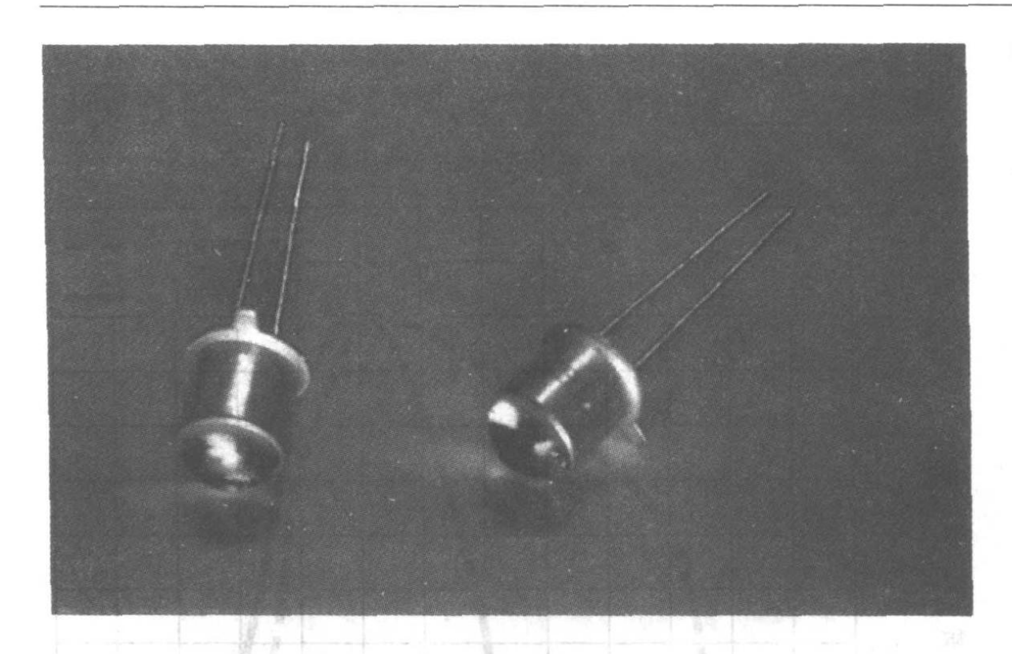


图 5.10　光电三极管实物照片

和光电二极管一样，在自制机器人中也很少单独使用光电三极管，它通常与发光二极管组合使用，如用在相扑机器人中检测场地四周的白色边界线等。光电三极管的结构及符号如图 5.11 所示。

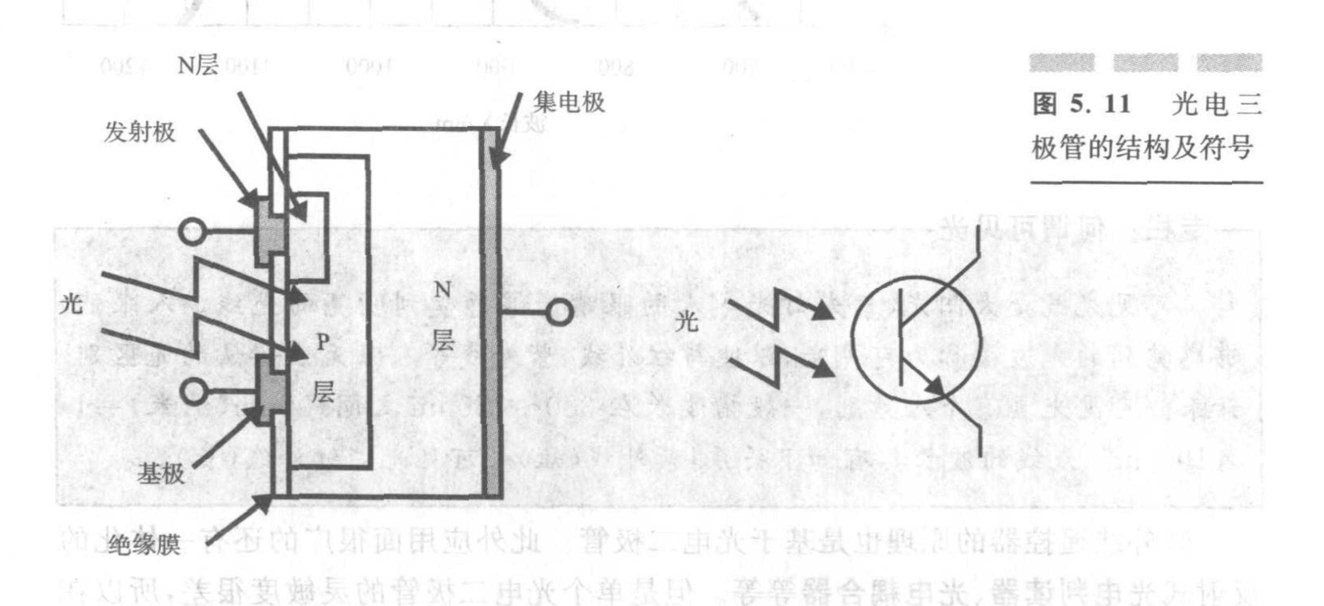
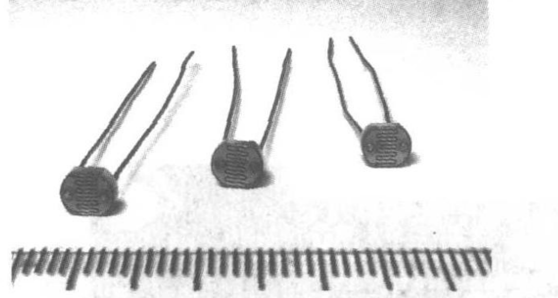


图 5.11　光电三极管的结构及符号

通常光电三极管的放大系数由电流放大率(h_{fe})决定，另外有一类光电达林顿晶体管有更高的放大系数，因此灵敏度得到提高。和前面介绍的光电二极管类似，光电三极管也可以用在光电判读器、光电耦合器当中，此时它们也多与发光二极管集成应用。

5.1.4 Cds 单元

Cds 单元是一种将硫化镉烧结在陶瓷基板上形成的半导体化合物，在光的照射下能与光强成正比地产生自由电子，从而产生电流，电阻阻值也随之变化。Cds 单元的实物如图 5.12 所示。



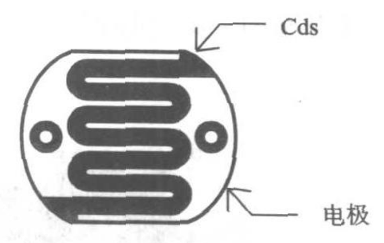


图 5.12 Cds 单元及其结构

Cds 单元随光线强弱引起的电阻阻值变化十分明显，但是变化存在较严重的时延现象，因此往往只用于缓慢变化的检测。在自制机器人中 **Cds 单元**的应用不太多，有时在无极性交流电路中见到，或者用于自动路灯开关器、照相机的曝光表等，在这些场合发挥它的灵敏度与人眼类似的特性。

5.1.5 图像传感器

图像传感器是一种将投影到传感器上的二维光图像转换成电信号的器件，典型的图像传感器分为 CMOS 和 CCD 两种，**CCD 图像传感器**的实物如图 5.13 所示。

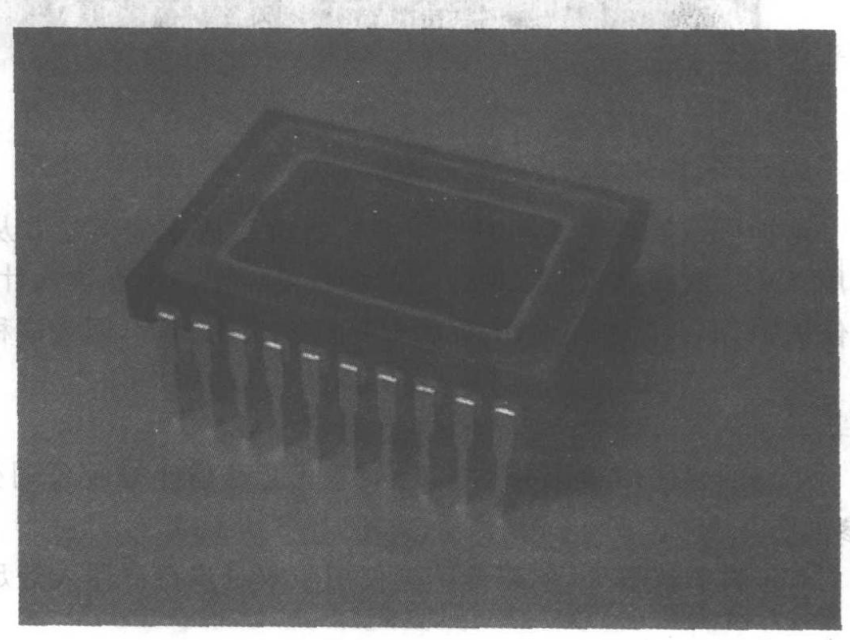


图 5.13 CCD 图像传感器实物照片（日本电气株式会社 PD3520D 型）



应用这种传感器产品的机器极其广泛，如人们熟悉的数字摄像机、可摄像手提电话、安全系统监控摄像头等，在医疗器械的内窥镜、地球卫星太空望远镜等也有应用。图像传感器产品如果像素较少，则不但价格便宜且很容易买到。

CCD、**CMOS 图像传感器**的工作原理如出一辙，无外乎借助二维分布的图像二极管将入射光线的强弱转换成电荷信号，并按照时间顺序提取电压信号。

在自制机器人中，往往采用图5.14的摄像处理单元，它集成了图像传感器、读取电路、同步电路等。

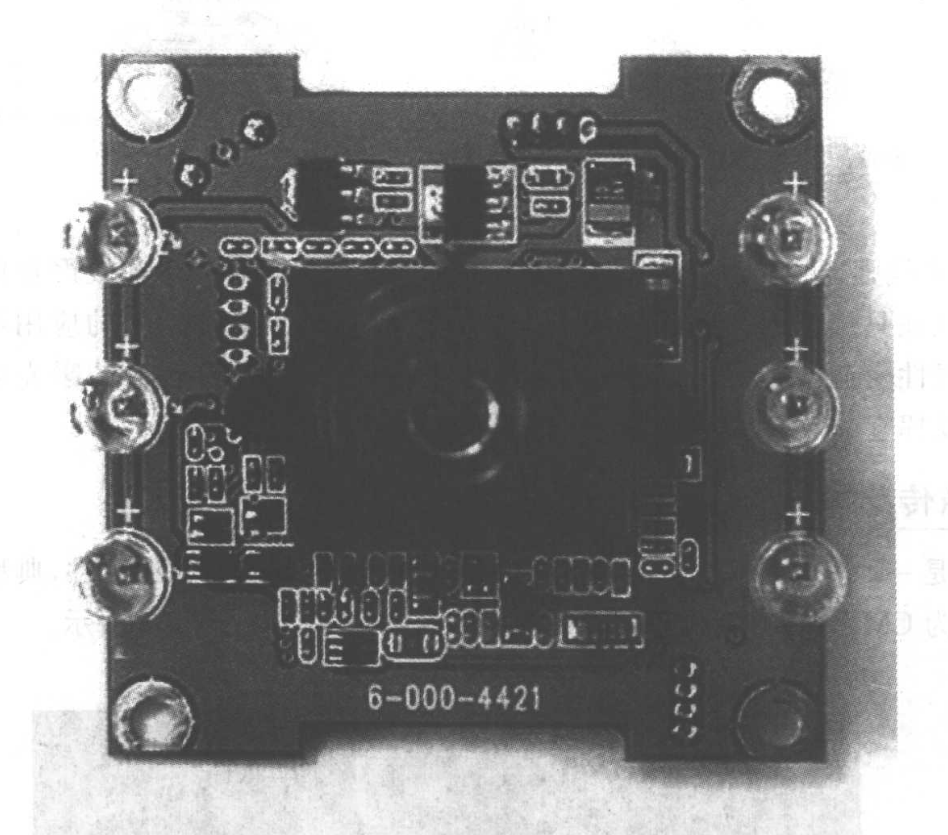


图 5.14　CCD 摄像头实物照片

这种摄像处理单元能够将图像信号转换成影像信号并从无线电波发射出去，再用TV接收和显示。如果应用的场合是将图像数据读入计算机进行物体辨识或图像识别，那么微控制器用的USB摄像头值得推荐，因为这种方案既便宜又简便。

5.1.6　感压传感器

小型的感压传感器并不多见。这里仅对稻叶橡胶公司生产的产品予以说明（参见图5.15）。

对于自制机器人，感压传感器仅用于双足步行机器人等足底压力检测等场合。

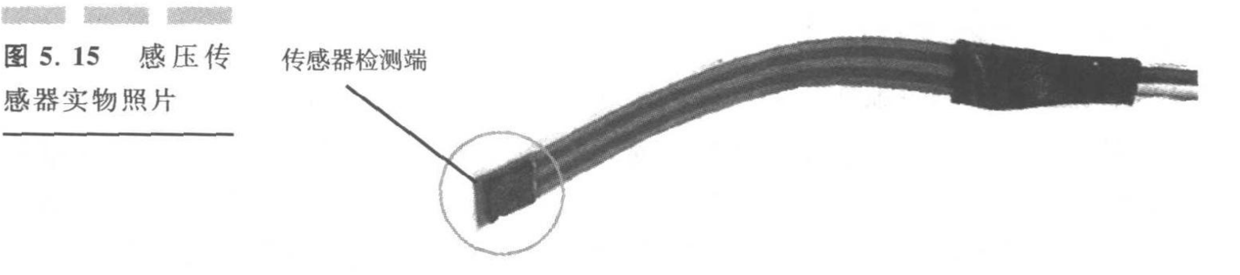


图 5.15　感压传感器实物照片

这种产品利用导电橡胶材料在压力作用下发生电阻变化的原理，反过来制成检测压力的感压传感器。如果将它们安装在足底的四个角，机器人就可以靠检测 4 个传感器的压力差来取得平衡。另外也可以根据传感器的阻抗特性，将传感器电流对应的电压信号做 A/D 转换（模拟量/数字量转换器），然后直接读取足底压力值的大小。

5.1.7　陀螺仪

陀螺仪是检测物体角速度的传感器，常用于运动物体的位置检测和姿态控制。陀螺仪有机械式、光学式、流体式、振动式等多种形式。振动式陀螺仪向压电陶瓷施加交变电压产生振动，如果此时振动体有旋转角速度，则会产生**哥氏力**，这就是角速度传感器的原理。

在行驶车辆的导航控制中，常常将车速传感器与陀螺仪组合使用，前者计算行进距离，后者计算方向。陀螺仪电路板的实物如图 5.16 所示。

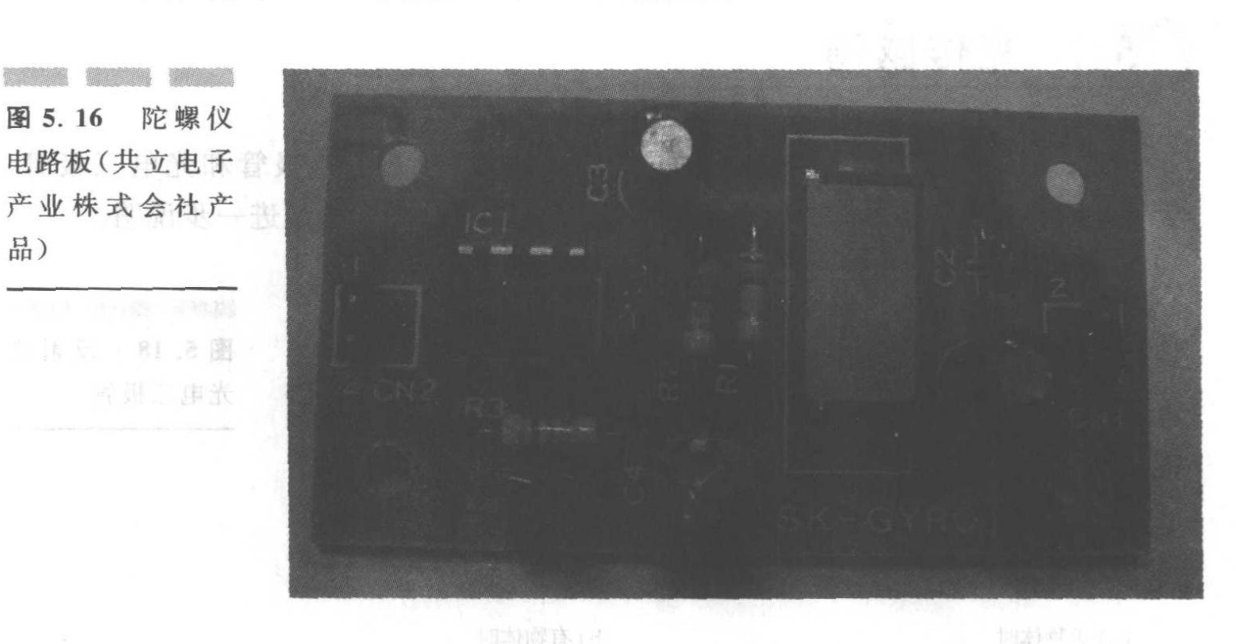


图 5.16　陀螺仪电路板（共立电子产业株式会社产品）

在自制机器人中，常用陀螺仪检测角速度实现双足机器人的姿态控制。

5.1.8 加速度传感器

加速度传感器能够测定加速度的变化，例如用在安全气囊、行驶车辆导航系统中。安全气囊的原理是发生车辆碰撞时，通过加速度的测定判断碰撞强度，并控制气囊弹出与否。加速度传感器的实物照片如图 5.17 所示。这款加速度传感器 ADXL202 在应变片做成的敏感元件周围嵌入电路，所以能直接输出数字信号。

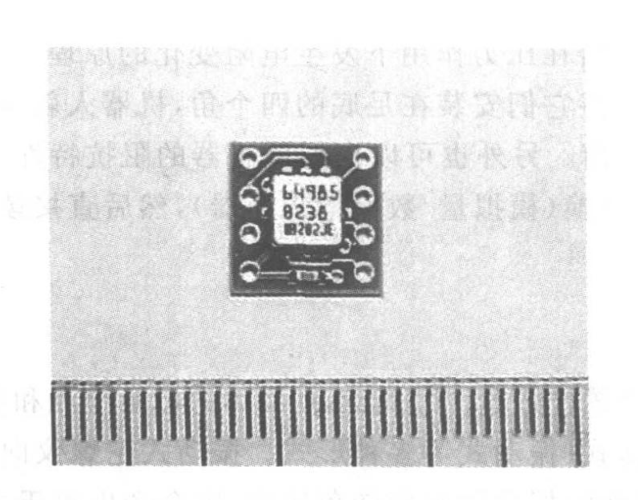
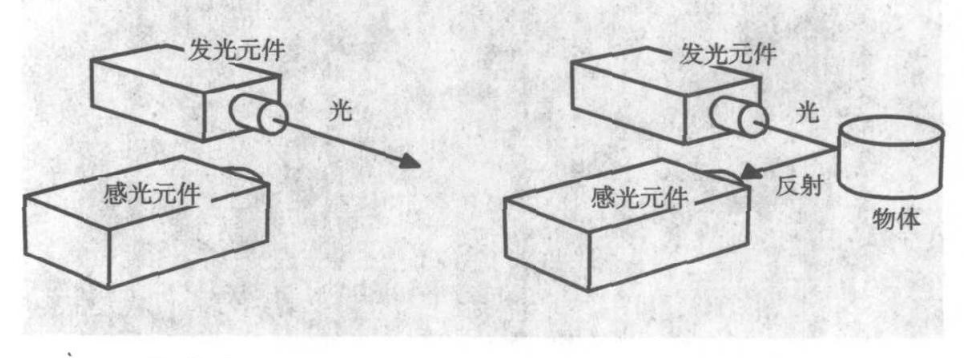


图 5.17 加速度传感器照片（ADXL 202：Analog Devices 公司产品）

在自制机器人中，加速度传感器与陀螺仪一样常用于双足步行机器人的姿态控制。类似的传感器还有 G 传感器，它测定包含重力加速度在内的加速度量。

5.2 光传感器

本章第 1 节已经对光传感器做了一些说明，例如光电二极管和光电三极管。本节就自制机器人常用的各种光电判读器的原理及使用方法做进一步说明。



(a) 无物体时　　(b) 有物体时

图 5.18 反射式光电三极管

按检测方法分类，光电判读器可分为图 5.18 的反射式、图 5.19 的透过式两种类型。

图 5.18(a)或图 5.18(b)的反射式光电判读器，其工作原理是从发光元件(如发光二极管等)发出的光被物体反射后到达感光元件，由此判断物体是否存在。显然，如果感光元件检测到反射光，表明前方有物体存在。

图 5.19(a)或图 5.19(b)的透过式光电判读器，其工作原理是从发光元件(如发光二极管等)发出的光被物体反射后是否到达感光元件，由此判断物体是否存在。与反射式光电判读器的区别在于，如果感光元件未检测到光线，则表明有物体存在。自制机器人一般采用反射式光电判读器，所以本节计划对它做重点说明。反射式光电判读器的实物及模型结构如图 5.20 所示。

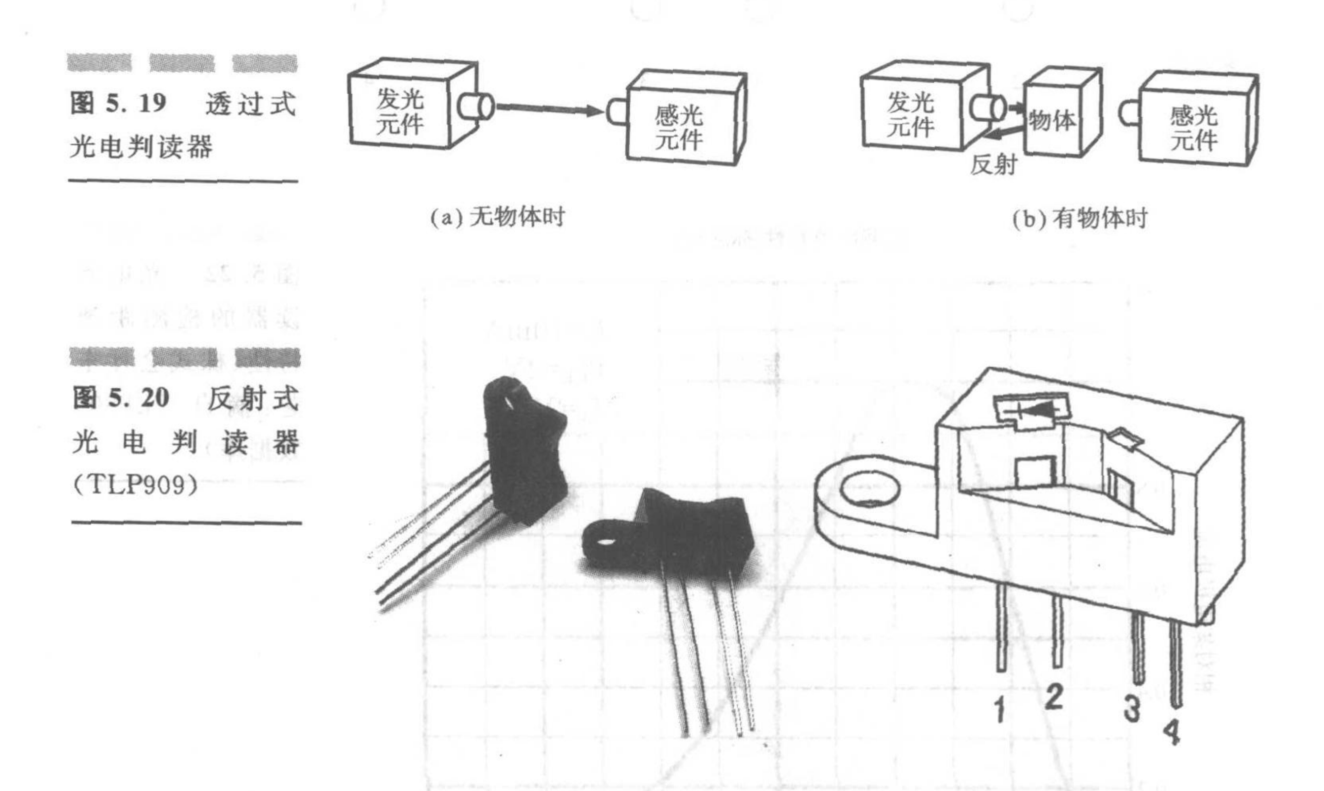


图 5.19 透过式光电判读器

图 5.20 反射式光电判读器(TLP909)

反射式光电判读器的内部电路如图 5.21 所示。除了东芝 Semiconductor 公司生产的 TPL909 之外，经常使用的同类传感器还有欧姆龙公司的 EE-SB5(B)和 LOOM 公司的 RPR-220 等。这些产品内部都有可见光隔离或过滤器，其作用是克服外界散乱光线干扰引发的误动作，所以它们的性能都比较稳定。图 5.22 为某个产品的检测距离特性图，从图中可知检测距离为 4mm 时输出最大。

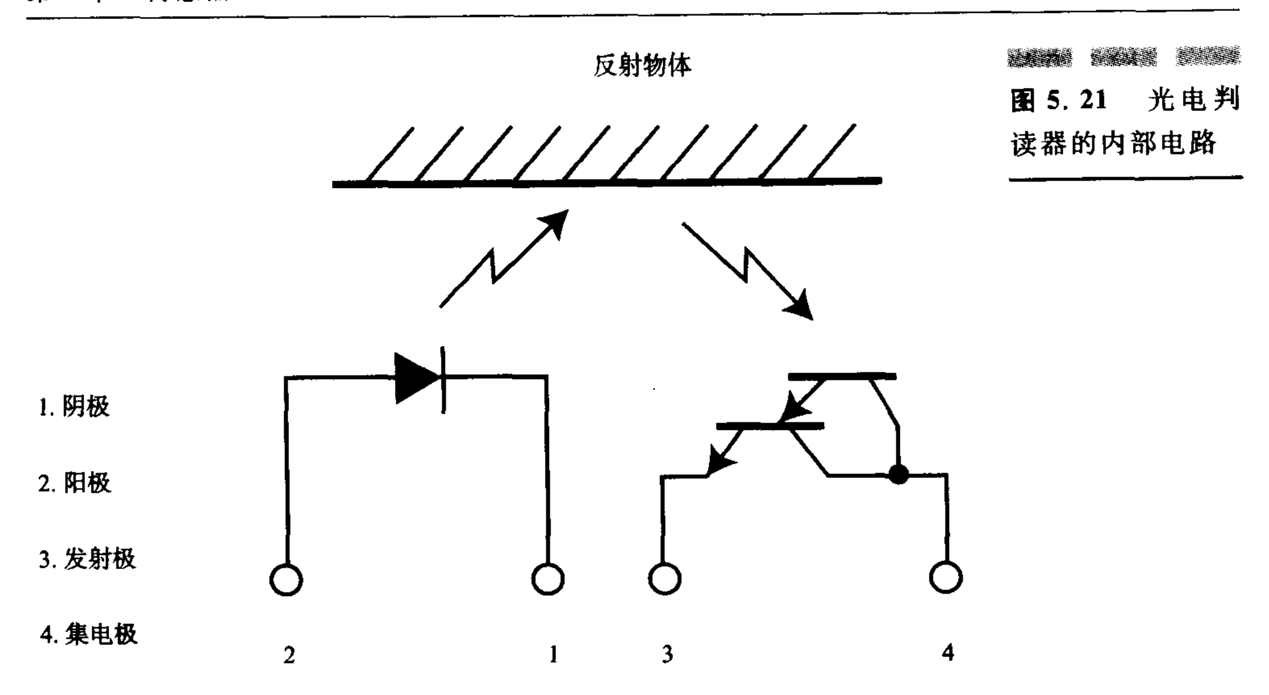


图 5.21　光电判读器的内部电路

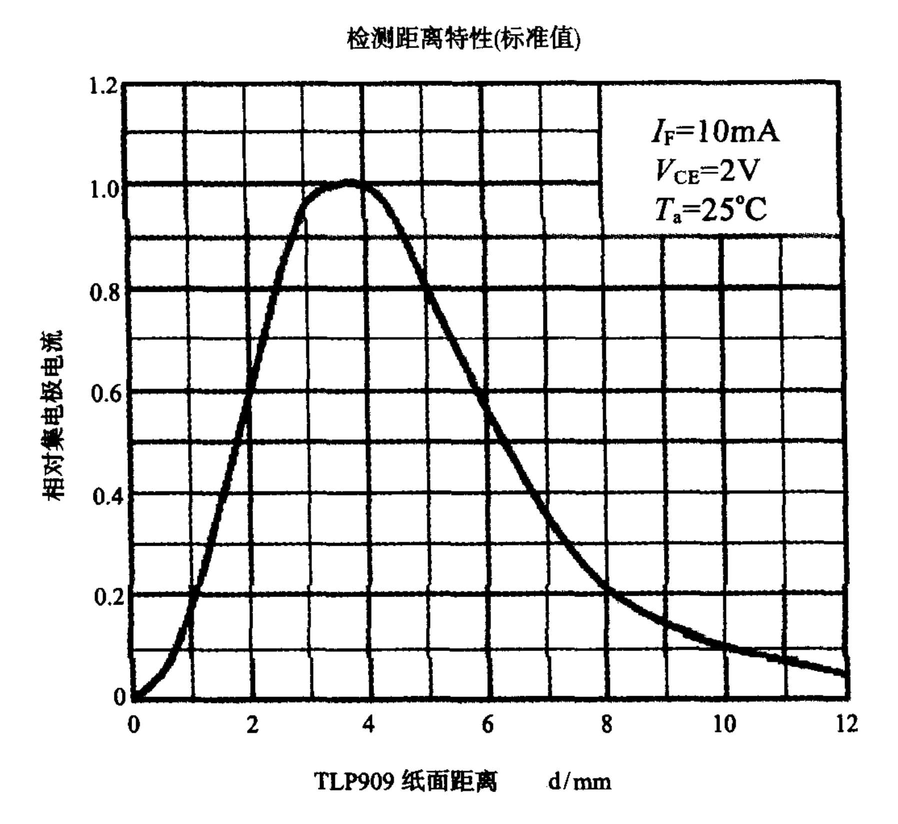


图 5.22　光电判读器的检测距离特性(株式会社东芝：摘自 TLP909 数据库)

自制机器人主要用反射式光电判读器。巡线小车上反射式光电判读器的安装方式如图 5.23 所示。

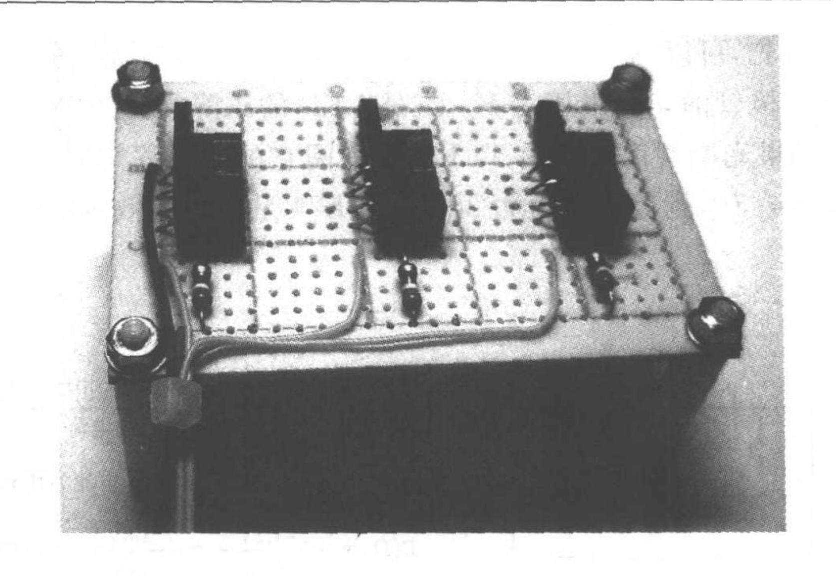


图 5.23 反射式光电判读器的安装

下面就第 7 章巡线小车的黑线检测电路做一些具体的说明(参见图 5.24)。

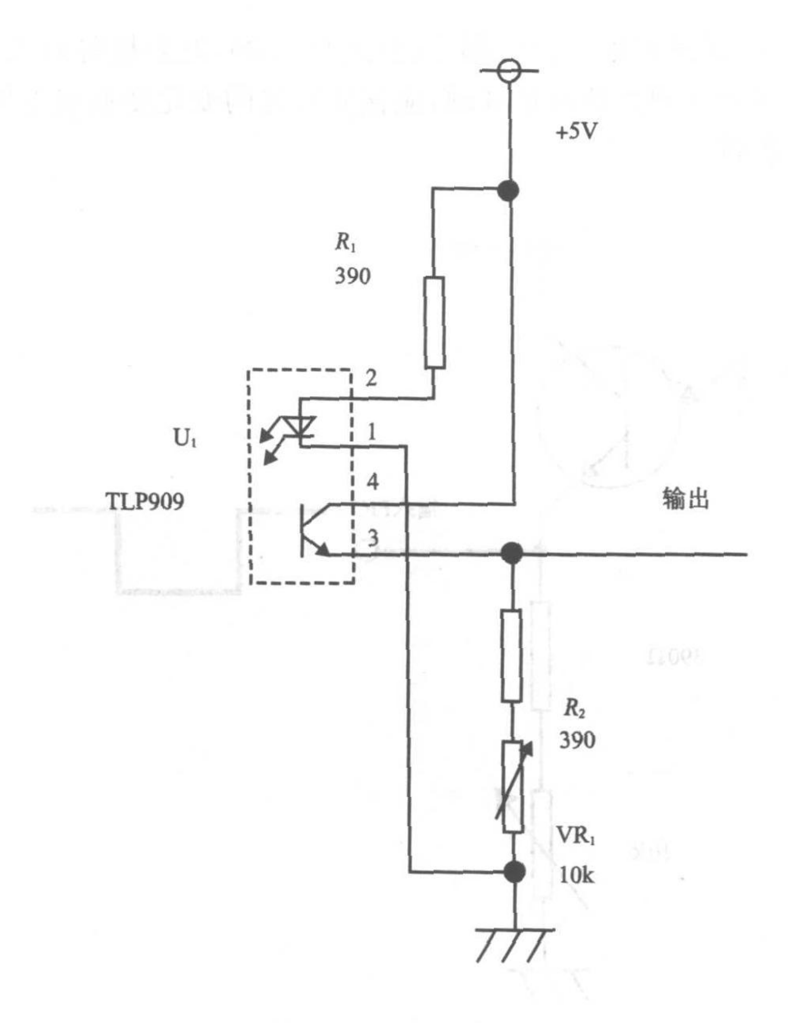
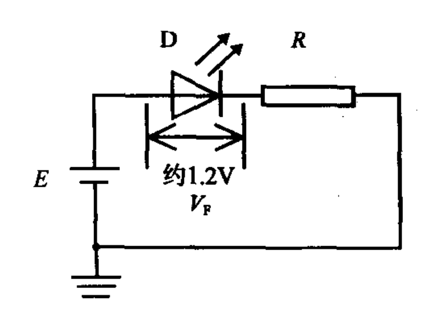


图 5.24 反射式光电判读器的检测电路

接下来讨论如何计算该电路中内置发光二极管的电流值,以及外加电阻的阻值(它是内置三极管的负载)。

首先,可以利用图 5.25 所示的计算公式求得发光二极管限流电阻的阻值。设发光二极管的正向电压为 V_F,红外 LED 的压降约为 1.2V,红色 LED 的压降约为 2V,如果光电判读器采用 TLP909,查阅它的数据库可知 V_F 为 1.2V,如果电流 I_F 按 10mA 计,则等效电阻为 380Ω,而 E24 系列的等效电阻可以取 390Ω。



$$R(\Omega)=\frac{\text{电源电压}E-\text{LED的正向电压}}{\text{LED的正向电流}}$$

*计算例,设电源电压为5V,电流为10mA,

$$R(\Omega)=\frac{5-1.2\text{V}}{10\text{mA}}=\frac{3.8\text{V}}{10\text{mA}}=380\Omega$$

图 5.25　发光二极管限流电阻的计算方法

其次,讲解光电三极管部分(参见图 5.26)发射极等效电阻阻值的计算。为了判别黑线及其周边的白色区域,应该先将光的变化变换成电压,然后输入 PIC 之类的微控制器。

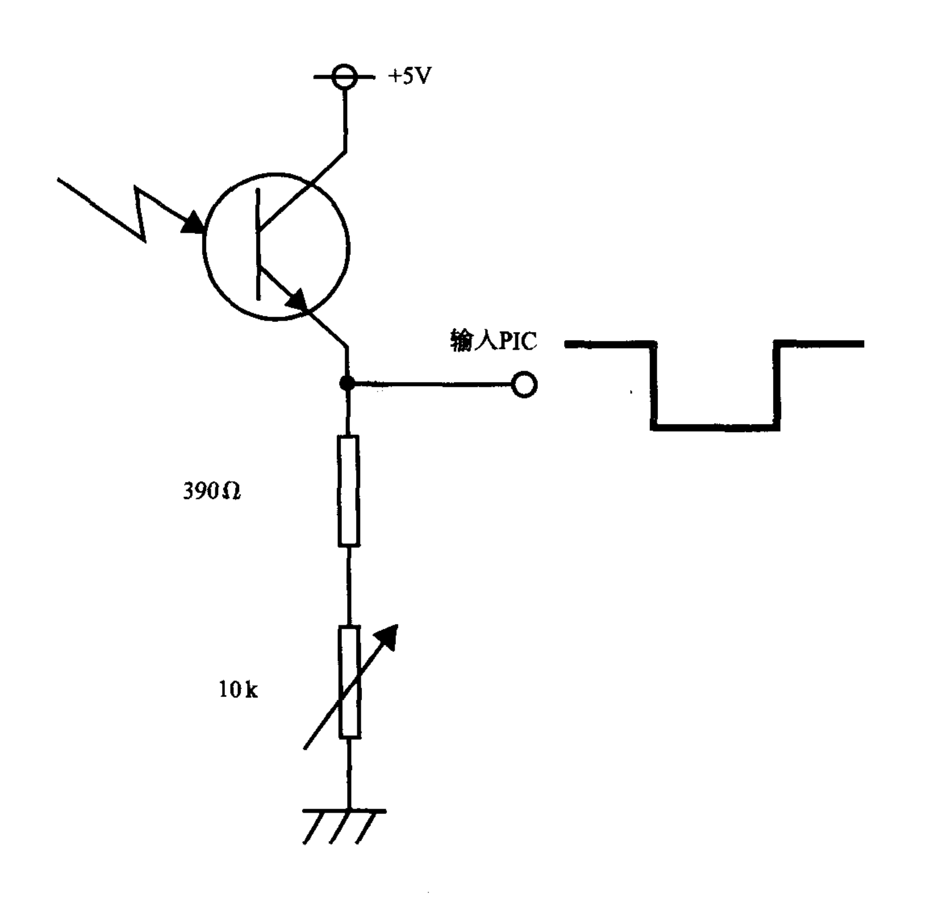
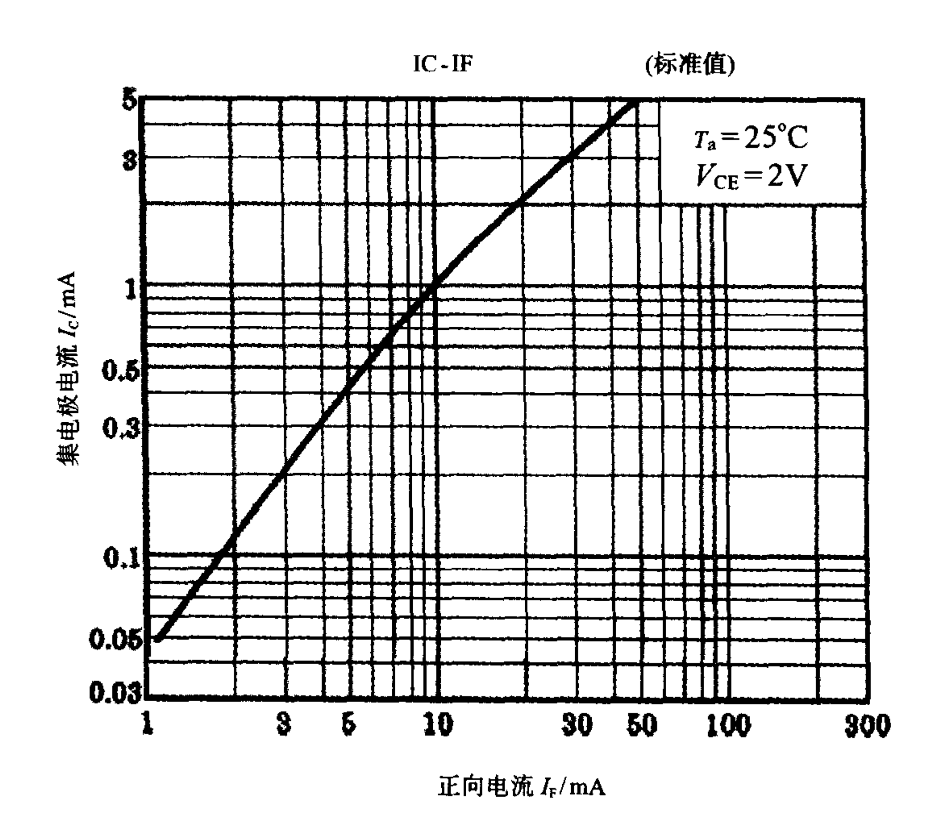


图 5.26　光电三极管部分的电路原理图

根据动作原理,检测到黑线时无反射光,对于光电判读器而言就是未感受到

光，因此没有基极电流发生，向 PIC 输出"0"信号。相反，在黑线以外的区域，则输出"1"信号。如果信号的输入对象为 PIC16F84A，查阅数据库得知它的"0"电平为 0.8V，"1"电平为 2.0V，根据这一输出电压就能够计算发射极的负载电阻。

图 5.27 I_C-I_F 特性曲线（TLP909）（摘自株式会社东芝数据库）



假设 I_F 为 10mA，反射光线充足，从图 5.27 的特性曲线可查阅出 $V_{CE}=2V$，$I_C=1mA$，于是输出电压为 5－2＝3(V)，超过了"1"电平的理论值 2.0V。因此可以按下式计算等效电阻。

$$R(\Omega)=\frac{5-2V}{1mA}=\frac{3V}{1mA}=3k\Omega$$

上述结果是在反射率良好的条件下求得的，I_C、V_{CE} 的实际值往往要低一些，所以建议等效电阻的实际阻值选得稍大一点。

也许读者会发问，光电判读器检测到黑线时发射极没有电流，那么这个等效电阻的阻值不就有问题了吗？其实，即使没有任何入射光线，光电判读器本身也存在漂移电流和漏电流，再加上外部散光源的干扰电流，在最坏的情况下，其值估计达到正常值的十分之一，即 0.1mA。根据上面所求得的等效电阻值计算，3kΩ×0.1 mA＝0.3V，这一数值比理论"0"电平要低，所以不会有任何问题。

在实际应用场合，显然黑线和白色区域的反射率都会随场地不同而异，因此发

射极的等效电阻可以用可变电阻来代替。本书第 7 章制作的巡线小车就采用了 10kΩ 的可调电阻，目的在于灵活地适应环境进行调整。图 5.24 中采取固定电阻加可调电阻串联的配置形式，固定电阻起发射极限流的作用，所以允许可调电阻的阻值调得很低。

5.3 超声波传感器

在相扑机器人之类的格斗型机器人中，超声波传感器是经常采用的传感器之一，用来检测对方机器人的有无和距离。其原理有如蝙蝠，它的嘴发出超声波，当超声波遇到小昆虫的时候，蝙蝠的耳朵能够接收反射回波，从而判断饵食的位置和距离并予以捕杀。超声波传感器的工作方式是通过发送器（相当于蝙蝠的“嘴”）发射出来的超声波被物体反射后传到接收器（相当于“耳朵”）接收来判断是否检测到物体（参见图 5.28）。

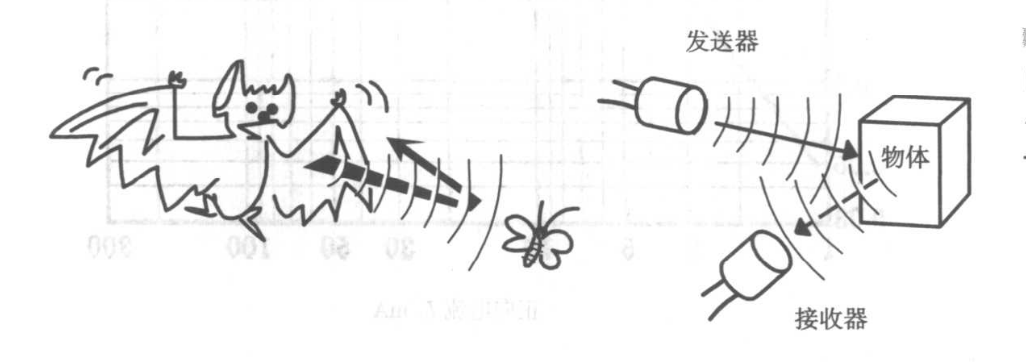


图 5.28　超声波传感器的工作方式

超声波传感器的实物照片如图 5.29 所示。

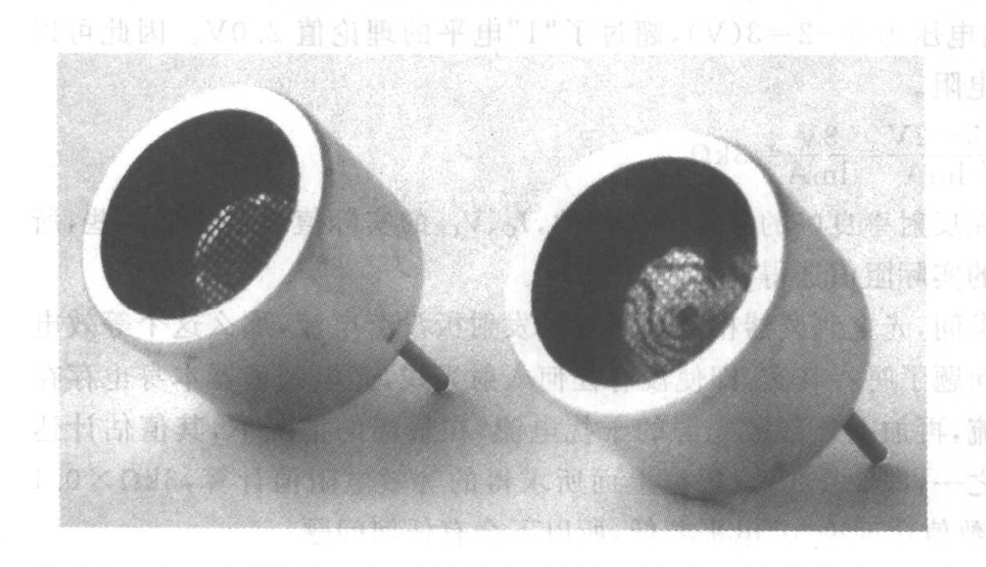


图 5.29　超声波传感器

人的听觉所能感觉的声音的频率范围往往因人而异，大约为 20Hz～20kHz。所谓超声波，即空气中传播的超过人类听觉频率极限的声波。kHz(千赫)代表一个频率单位，20kHz 即在 1 秒钟内来回振动 2 万次的频率状态。众所周知，超声波的传播速度 V 可以用下式表示：

$$V=331.5+0.6T(\mathrm{m/s})$$

式中，T(℃)为环境温度，在 23℃的常温下超声波的传播速度为 345.3m/s。超声波传感器一般就是利用这样的超声波来检测物体的。

具体到超声波传感器的结构，一般它的内部都有一个振子(参见图 5.30)，所谓振子是在一块金属片上贴着压电陶瓷。

图 5.30 超声波传感器的结构(摘自日本陶瓷株式会社)

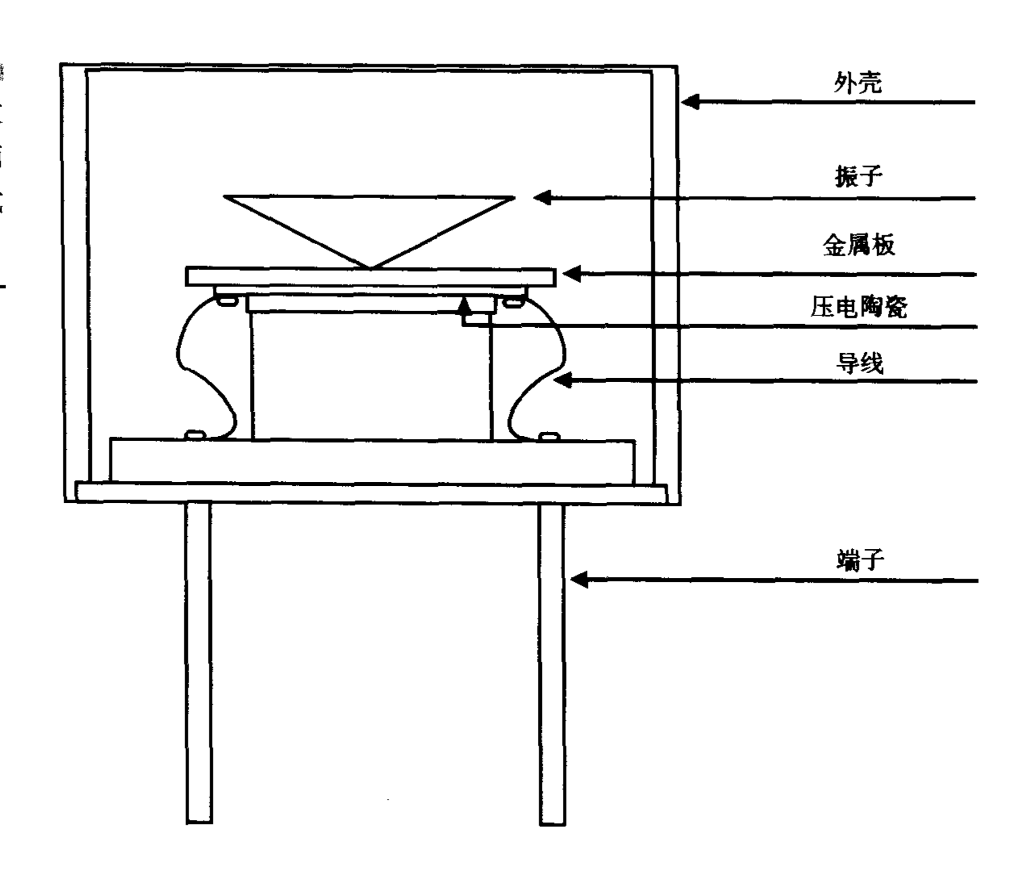


图 5.31 表示，如果给压电陶瓷加上电压，它就会根据电压的大小产生相应的机械变形，进而产生机械振动，这就是所谓的压电现象。超声波传感器就是利用这种原理实现超声波的发射与接收的。

与前面提到的光电判读器类似，超声波传感器也有透过式和反射式之分。超声波传感器不但可以检测物体，还可以通过发射器的回波返回接收器所需要的时间来测量距离。

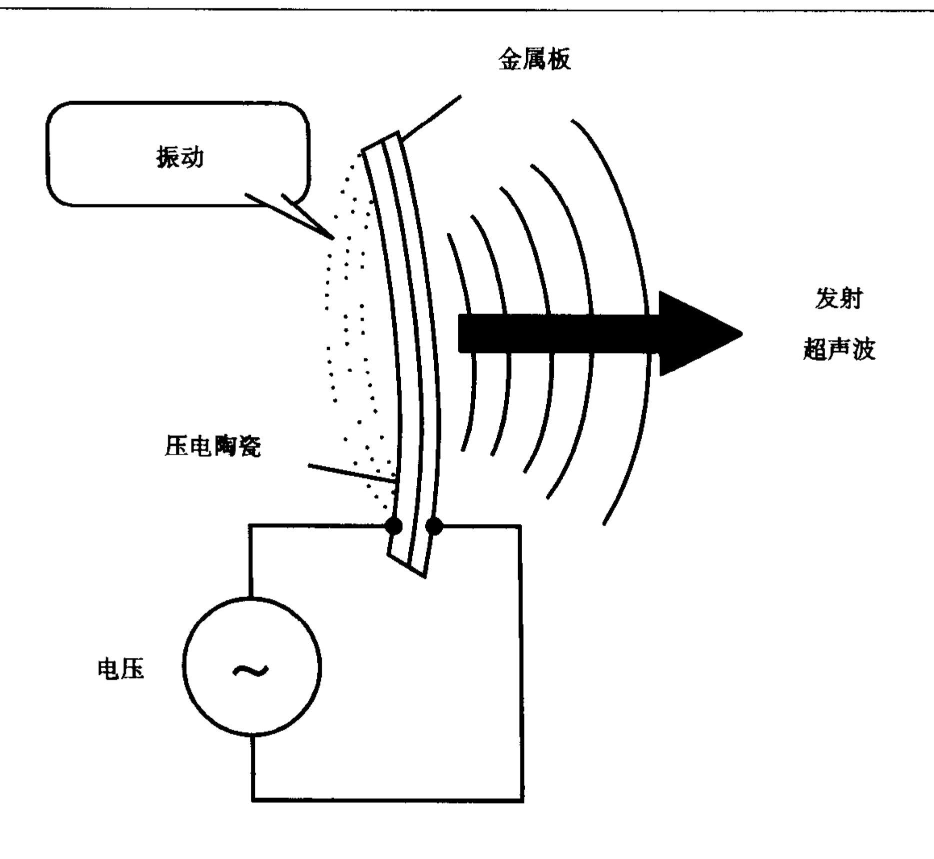


图 5.31　超声波传感器的工作原理

图 5.32 为某一个相扑机器人的超声波传感器应用电路，用于检测来自对方的反射波，从而判别对方所在的位置。

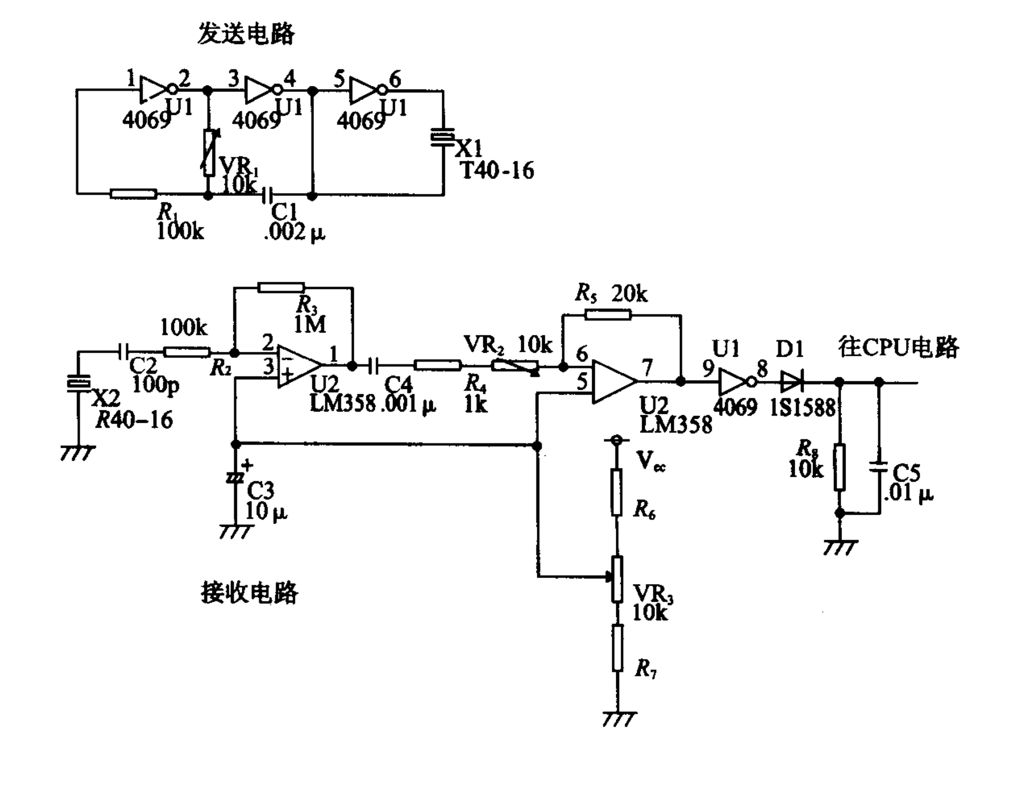
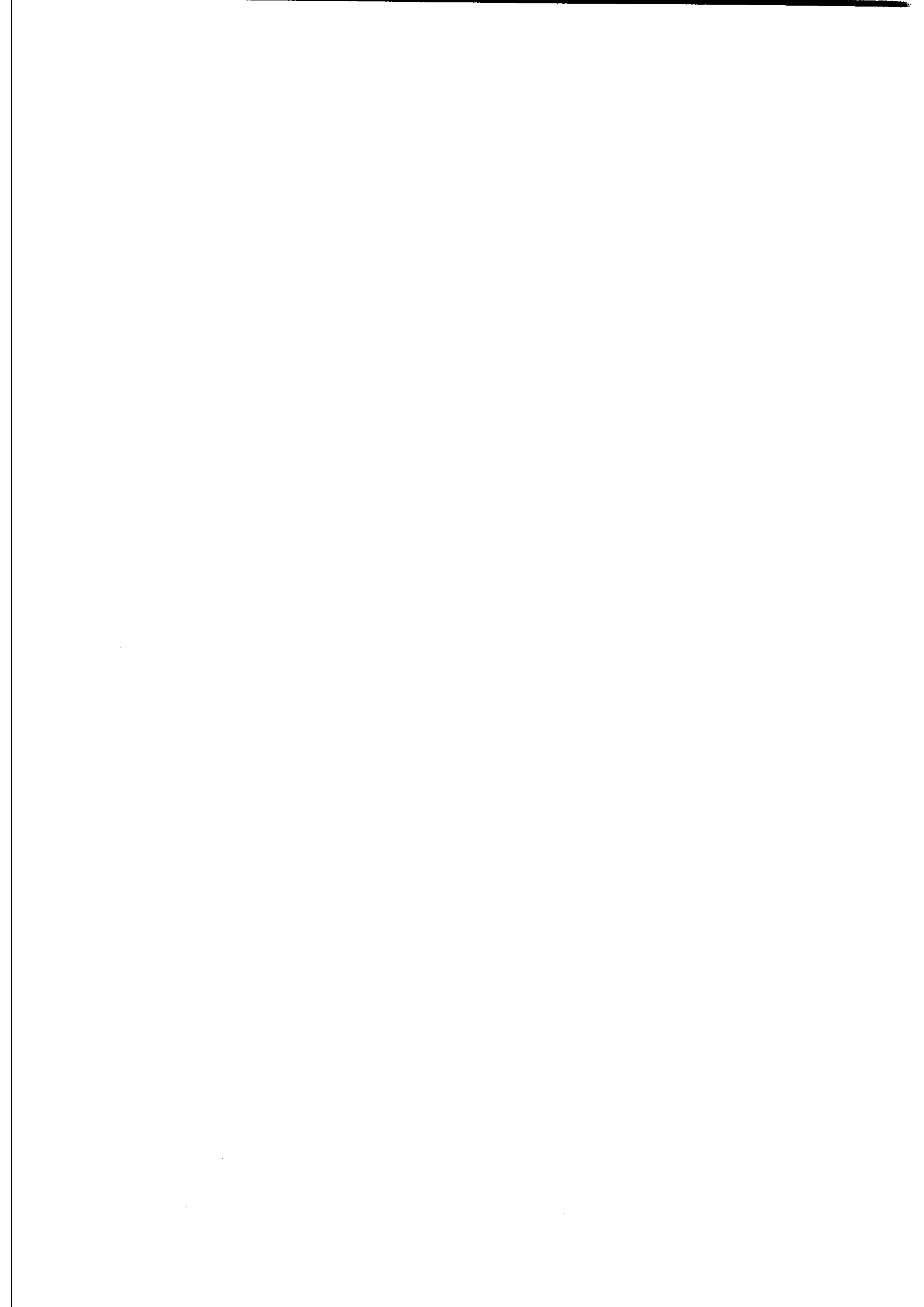


图 5.32　超声波传感器的电路图实例（摘自：电子控制入门，堀桂太郎著，OHM，1998）

练习题

1. 写出相当于人类五官感觉的传感器。
2. 说出可见光波长的范围。
3. 简要说明安全气囊装置中使用的传感器及其使用方法。
4. 写出可听声音的频率范围。
5. 计算环境温度为 0℃和 50℃时的声音传播速度。
6. 计算图 5.25 反射式判读器检测电路中采用红色 LED 时限流电阻的阻值。
7. 以反射式光电判读器为例,用图示的方法简要地说明反射检测物体的原理。



第6章

红外线遥控装置

人在远距离对机器人发送指令称为遥控操作，遥控操作可分为有线方式和无线方式两类。从提高自主移动机器人的灵活性出发，采用无线方式更为便利。

无线遥控装置有无线电、声波、光等多种形式，本章重点学习红外线无线遥控装置。电视机和空调器的遥控器就是红外线的，其电路比较简单而且性能上佳。

本书第7章后半部分有关巡线小车的内容与本节的红外线遥控装置有密切的关系。

6.1 红外线遥控装置的工作原理

如图6.1所示，红外线是波长约大于770nm的不可见光。将发射红外线的发光二极管(LED)和感知红外线的接收模块组合在一起就可以实现红外线通信。

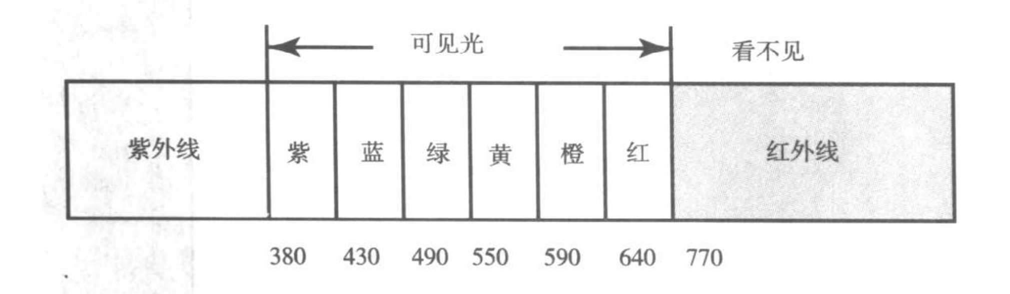
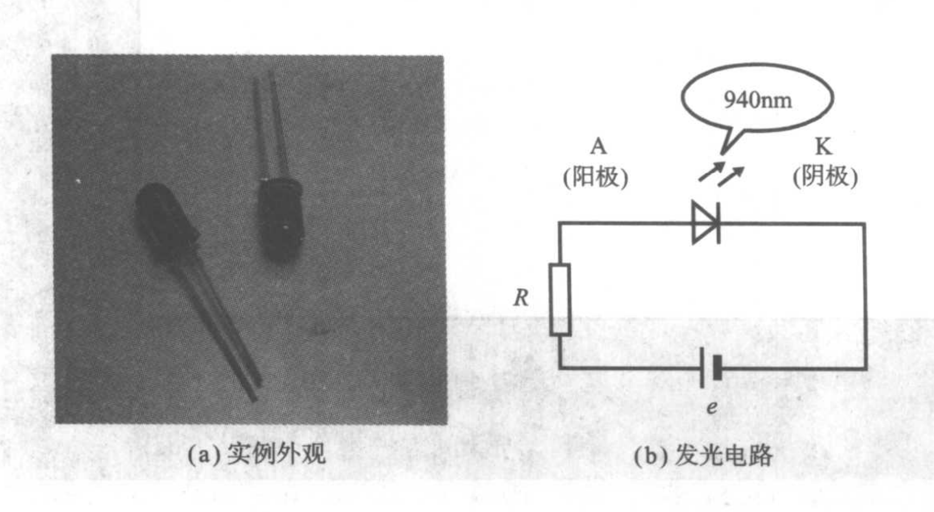


图6.1 光的颜色及波长

6.1.1 红外线LED和接收模块

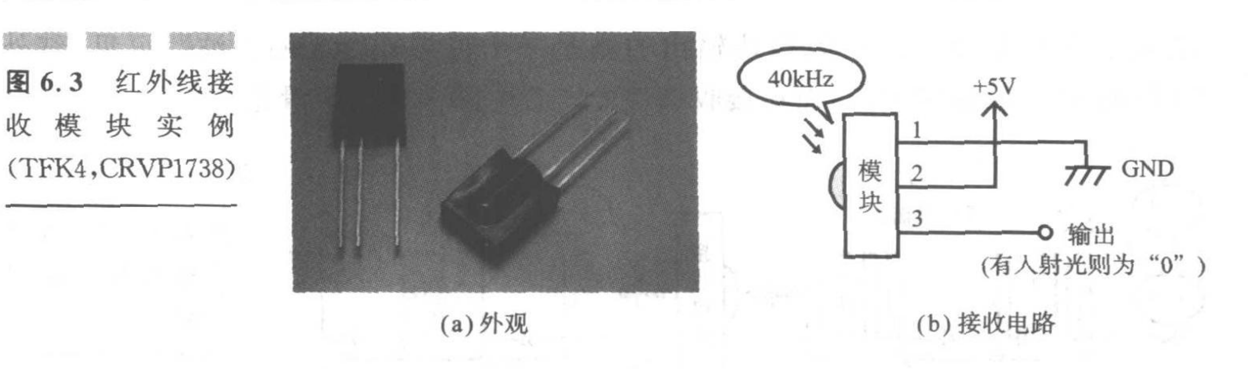
图6.2为红外线LED，图6.3为红外线接收模块。



(a) 实例外观 (b) 发光电路

图6.2 红外线LED的实例(TLN110:株式会社东芝)

图 6.3 红外线接收模块实例(TFK4,CRVP1738)



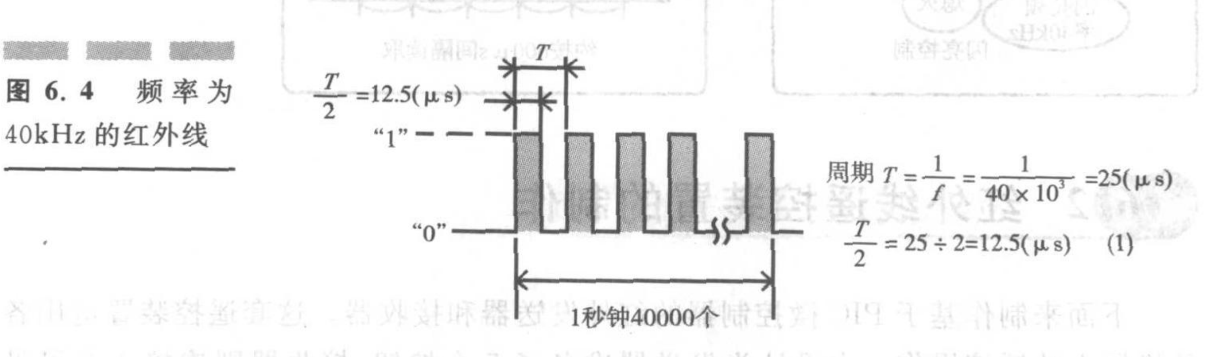
东芝公司生产的 TLN110 是红外线 LED,波长为 940nm,适合遥控应用。红外线 LED 的接通装置非常简单,如图 6.2(b)所示,与普通红色或绿色 LED 不同的是人眼无法分辨红外线 LED 是否点亮工作了。使用过程中应注意不要超过最大电流,TLN110 的直流正向电流的最大值为 100mA。TLN105B 也是一款与 TLN110 规格基本相同的红外线 LED。

红外线接收模块 TFK4(同类产品有 CRVP1738)由接收 PIN 二极管、电压控制电路、带通滤波电路、解调电路等组成的集成电子单元。在未接收到红外光线前它的输出为“1”,接收到红外光线后输出为“0”,这种现象被称为“低电平激活”。

6.1.2 基于 PPM 的红外线通信方式

应用红外线接收模块通信时,必须将所使用的红外线设置成图 6.4 所示的频率 40kHz(CRVP1738 的频率为 38kHz)的方波形式,即 1 秒钟内断续地发送红外线 4 万次,这种调制方式被称为脉冲周期调制(PPM)。根据(1)式可以计算出 1 个方波中为“1”的时间为 12.5μs。

图 6.4 频率为 40kHz 的红外线



当向红外线接收模块输入“1”时信号是 12.5μs 的间隔,红外线 LED 则以 600μs 的持续时间闪亮(向外发送红外线);当输入信号“0”时,红外线 LED 持续停止工作 600μs。图 6.5 表示向接收模块输入长度为 3 位的信号“101”的实例,每 1 位的时间对应 600μs,这是从接收模块规格表中查到的标准值。为了适应低电平激活的特点应该将输入信号反转后输出。当输入信号为“1”时,每个输入脉冲都是

12.5μs 的方波，不过从接收模块输出的则是一个持续 600μs 的方波。可见，红外 LED 的 40kHz 频率闪亮，而在接收模块的内置电路中得到平滑化。

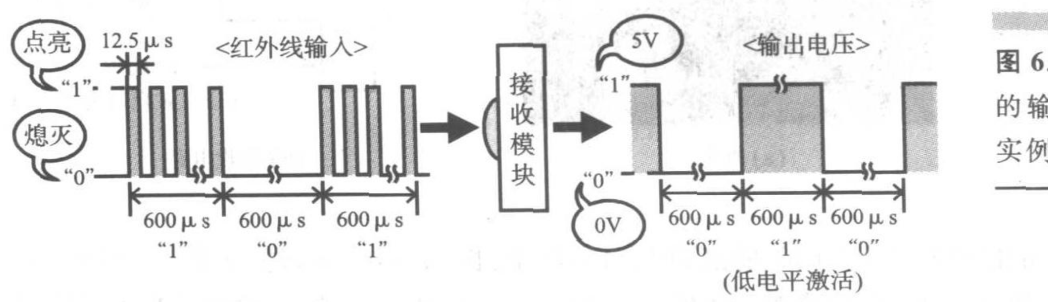


图 6.5 接收模块的输入输出信号实例

综上所述，在发送器一侧，应该根据被发送信号来控制红外线 LED 的点亮和熄灭；而接收器一侧，应该大约按照 600μs 的周期间隔读取接收模块输出的信号（参见图 6.6）。在这样的电路中最好采用 PIC 微控制器。

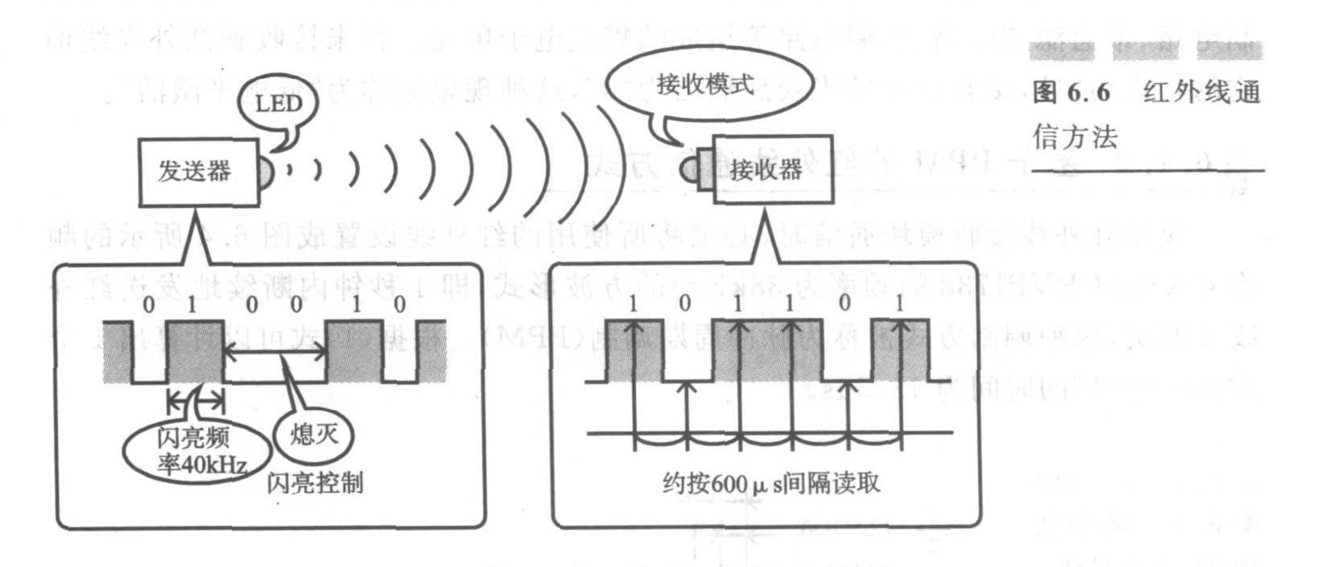
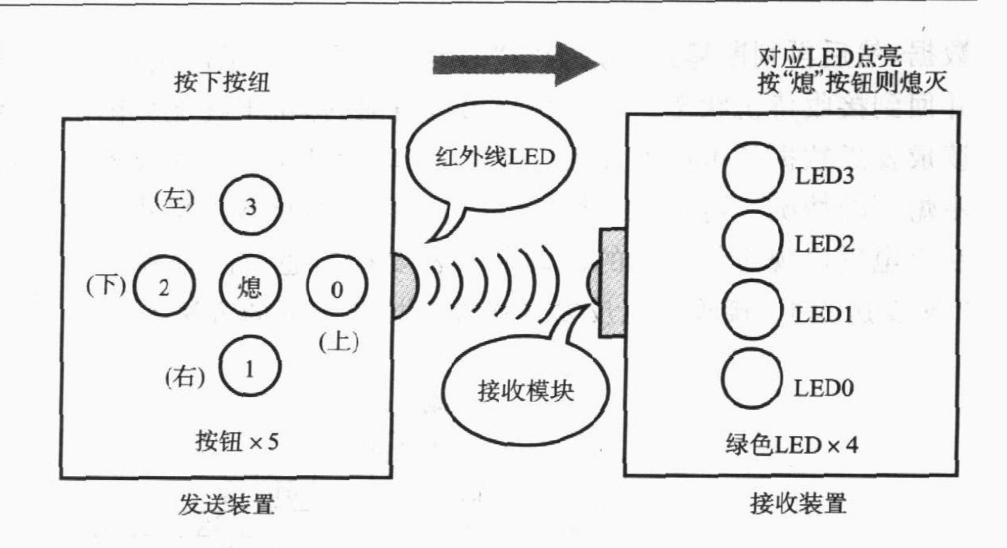


图 6.6 红外线通信方法

6.2 红外线遥控装置的制作

下面来制作基于 PIC 微控制器的红外发送器和接收器。这套遥控装置适用各种机器人的遥控操作。本设计为发送器准备了 5 个按钮，接收器则连接 4 个可见光 LED(参见图 6.7)。每个按钮对应控制一个 LED 点亮，中间的按钮则用来控制所有 LED 的熄灭。

图 6.7 红外线遥控装置框图

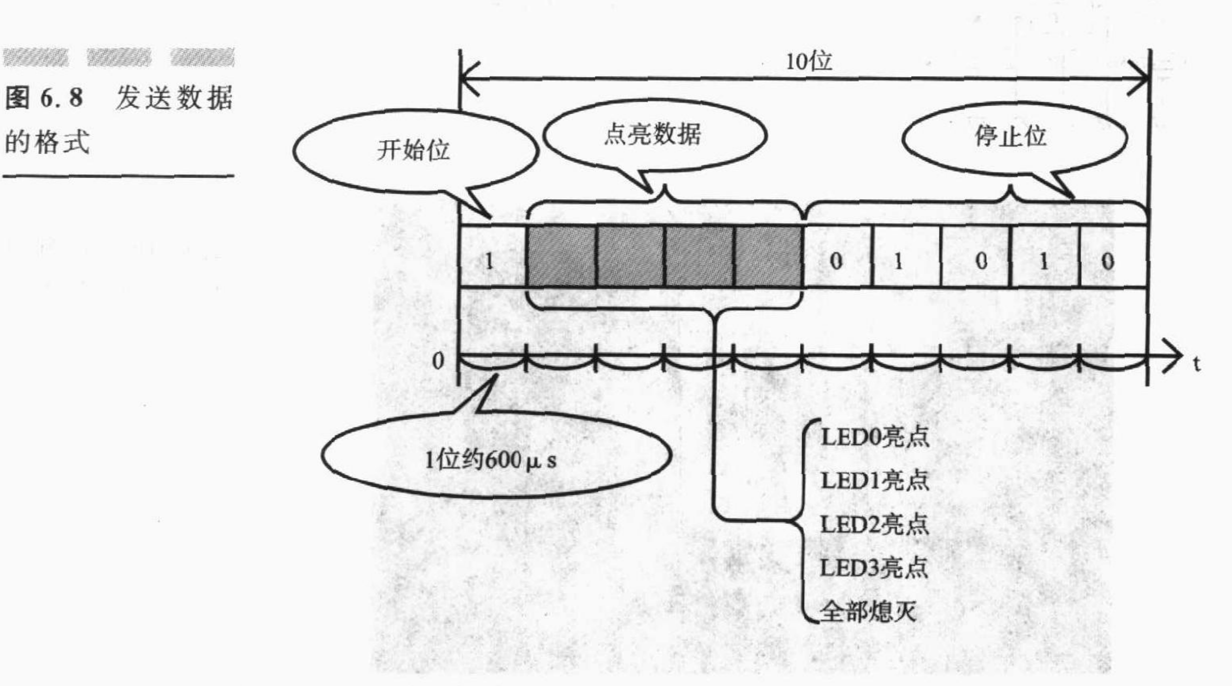


在红外线遥控装置的程序中，必须正确使用定时器，但本书所采用的 PICC Lite 不含有定时器函数，因此，本书中红外线遥控装置的程序需要用汇编语言来编制。

6.2.1 红外线遥控装置的发送器

图 6.8 表示发送数据的格式。

图 6.8 发送数据的格式



由图 6.8 可知，在发送器一侧，每次发送十位数据，第 1 位为开始位（"1"），其后的 4 位为 LED 的点亮数据，最后的 5 位为停止数据（"01010"），每一位的发送时间大约为 600μs。为了防止发送端的误动作，当发送完 1 次数据后设置了 20ms 的停止时间。

在接收器一侧，以发送数据的**开始位**为起始标志，从其后续的 4 位中取得亮灯

数据,然后再判断**停止位**是否与“01010”一致。如果停止位不符,则放弃亮灯数据,并回到接收待机状态。停止位也可以省略,但抗干扰能力随之下降,将干扰信号认读成发送数据的可能性加大。比如,实验时甚至出现接收器上电后 LED 始终点亮不熄灭的情况。可见,如果机器人遥控接收器将数据格式的停止位省略,同样可能在上电后出现意想不到的误动作。若要提高通信的可靠性,可以要求发送器重复 2 次发送数据,接收器也反复 2 次校对接收数据的方法。

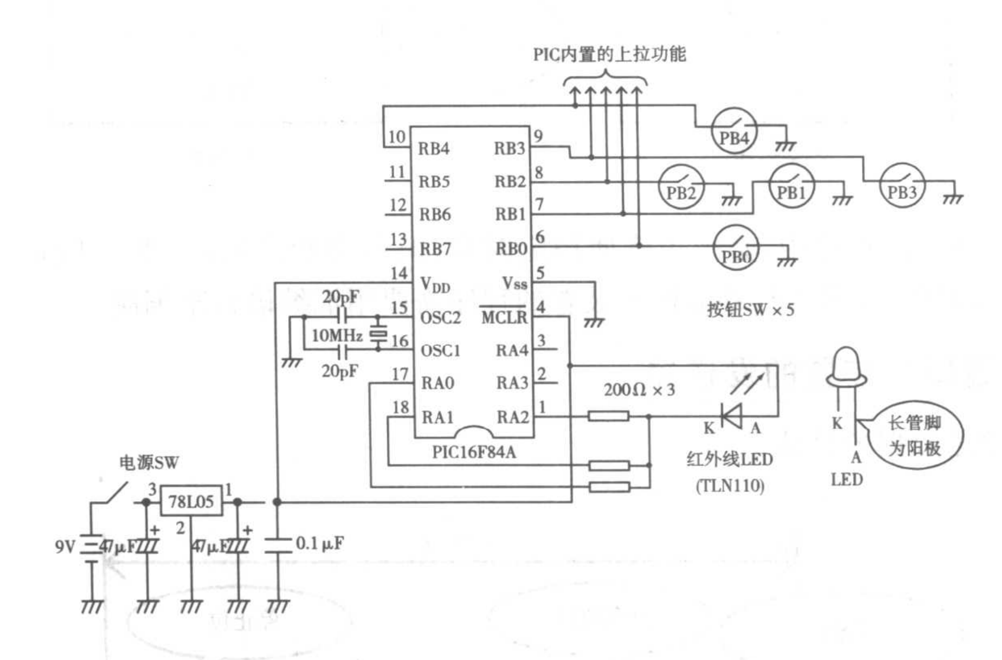


图 6.9　红外线遥控发送器的电路

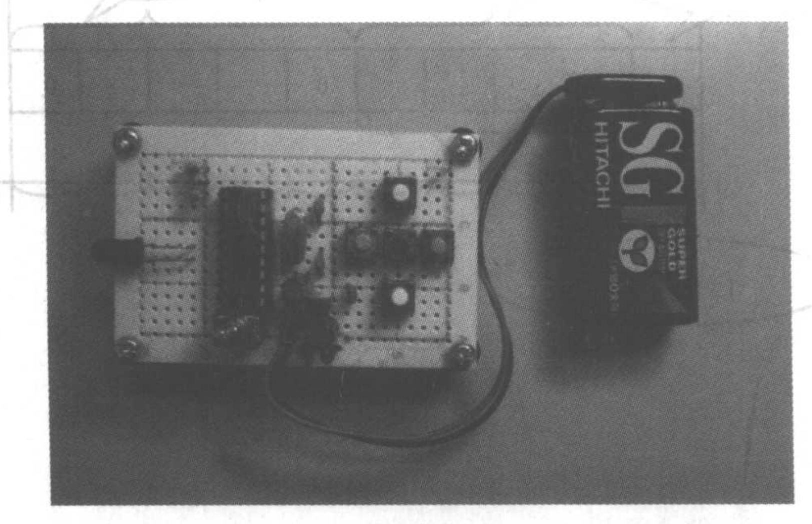
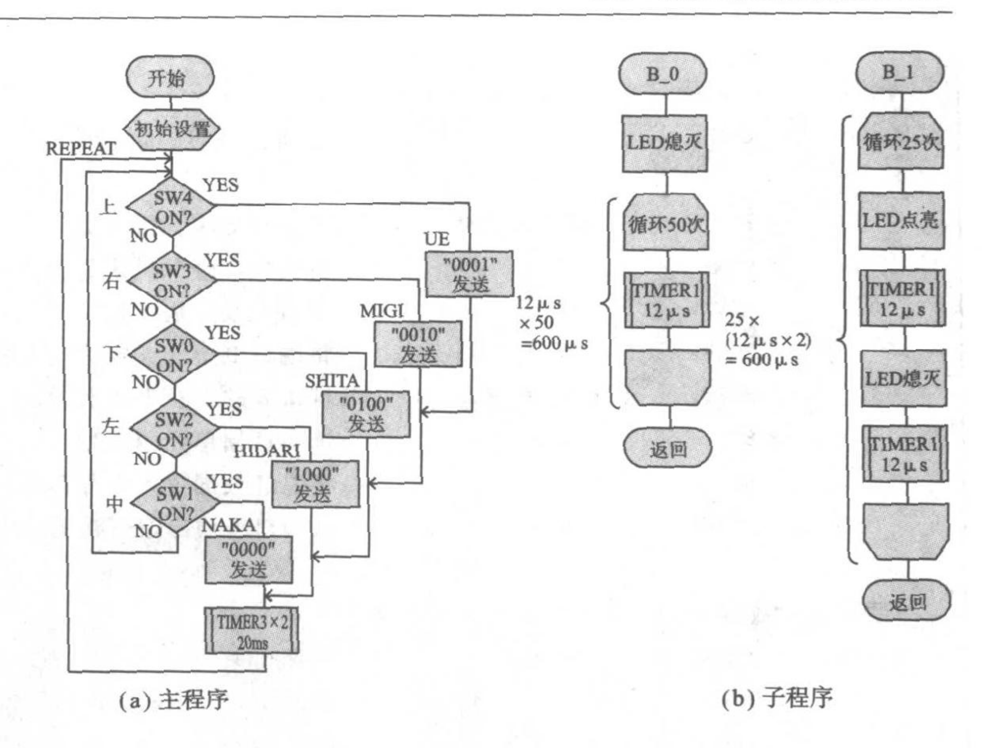


图 6.10　自制红外线遥控发送器的实物照片

图 6.9 为基于 PIC16F84A 微控制器的红外线遥控发送器的电路图,图 6.10 为制作好的实物照片。

它们的时钟振荡电路采用了 10MHz 的石英振荡器。调低振荡频率能降低功率消耗,但是必须重新计算时间子程序中的时间值。

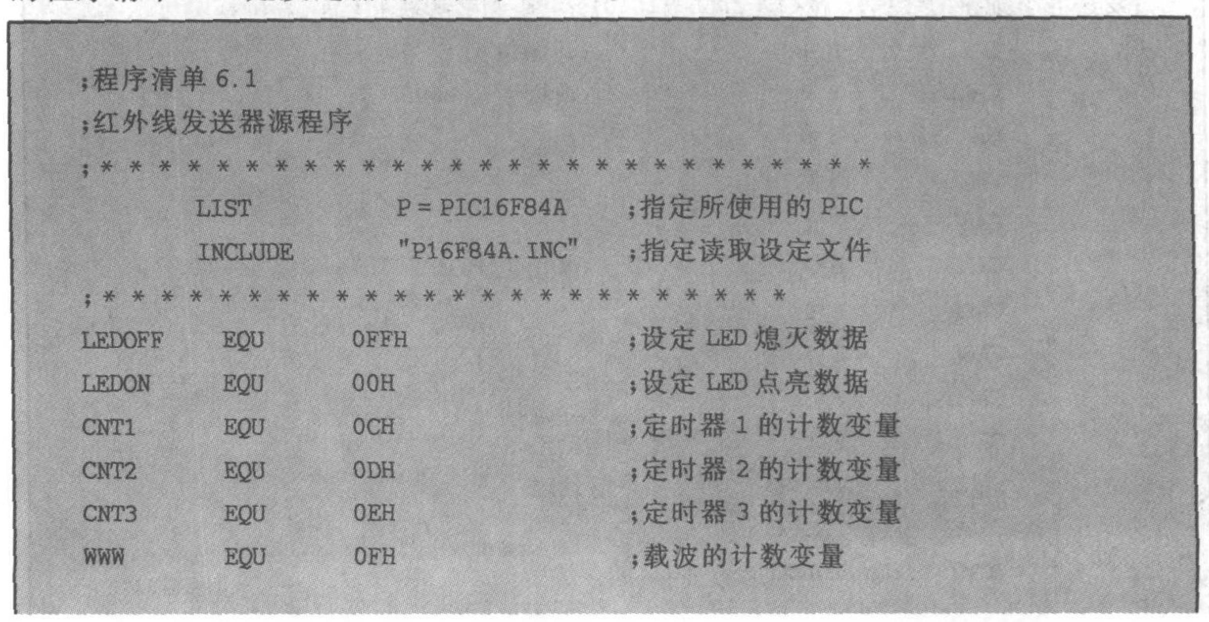
图 6.11 发送器程序框图



(a) 主程序　　(b) 子程序

图 6.11(a)为发送器的主程序方框图,图 6.11(b)为产生信号"0"和"1"的子程序方框图。

该程序逐次判断各个按钮,其间如果发现有被按下的开关,就在数据开始位和停止位之间插入对应于的点亮数据发送。发送 1 个"0"位,红外线 LED 熄灭大约 600μs 的间隔;发送 1 个"1"位,LED 按照 12.5μs 的间隔反复点亮和熄灭大约持续 600μs 的时间。由于这些时间间隔都是靠定时器子程序来设定的,会存在一定程度的误差,例如反复调用定时器子程序时,就未将调用的时间计算在内。下面给出的程序清单 6.1 是发送器的源程序。


```
;程序清单 6.1
;红外线发送器源程序
;*****************************
        LIST        P=PIC16F84A       ;指定所使用的 PIC
        INCLUDE     "P16F84A.INC"     ;指定读取设定文件
;*****************************
LEDOFF   EQU    0FFH                  ;设定 LED 熄灭数据
LEDON    EQU    00H                   ;设定 LED 点亮数据
CNT1     EQU    0CH                   ;定时器 1 的计数变量
CNT2     EQU    0DH                   ;定时器 2 的计数变量
CNT3     EQU    0EH                   ;定时器 3 的计数变量
WWW      EQU    0FH                   ;载波的计数变量
```

```
;*****************************
        ORG         0               ;存储程序的起始地址

        BSF         STATUS,RPO      ;选择存储单元第1列
        CLRF        TRISA           ;将端口A全部设定为输出模式
        MOVLW       0FFH            ;端口B设定用数据
        MOVWF       TRISB           ;将端口B全部设定为输入模式
        BCF         OPTION_REG.,7   ;端口B的上拉电阻有效
        BCF         STATUS,RPO      ;选择存储单元第0列
        MOVLW       LEDOFF          ;将熄灯数据置入W寄存器
        MOVWF       PORTA           ;从端口A输出熄灯数据

;读取按钮数据*******************
REPEAT  BTFSS       PORTB,4         ;检查SW4
        GOTO        UE              ;若为0(ON)则跳转至上
        BTFSS       PORTB,3         ;检查SW3
        GOTO        MIGI            ;若为0(ON)则跳转至右
        BTFSS       PORTB,0         ;检查SW0
        GOTO        SHITA           ;若为0(ON)则跳转至下
        BTFSS       PORTB,2         ;检查SW2
        GOTO        HIDARI          ;若为0(ON)则跳转至左
        BTFSS       PORTB,1         ;检查SW1
        GOTO        NAKA            ;若为0(ON)则跳转至中
        GOTO        REPEAT          ;重复

;发送各个按钮的数据,8位(1+ABCD+10101)*********
UE      CALL        B_1             ;开始位1
        CALL        B_0             ;数据4位B0001
        CALL        B_0
        CALL        B_0
        CALL        B_1
        CALL        B_1             ;停止位10101
        CALL        B_0
        CALL        B_1
        CALL        B_0
        CALL        B_1
        CALL        TIMER3          ;延时
        CALL        TIMER3
        GOTO        REPEAT
```

```
MIGI    CALL    B_1         ;开始位 1
        CALL    B_0         ;数据 4 位 B0010
        CALL    B_0
        CALL    B_1
        CALL    B_0
        CALL    B_1         ;停止位 10101
        CALL    B_0
        CALL    B_1
        CALL    B_0
        CALL    B_1
        CALL    TIMER3      ;延时
        CALL    TIMER3
        GOTO    REPEAT

SHITA   CALL    B_1         ;开始位 1
        CALL    B_0         ;数据 4 位 B0100
        CALL    B_1
        CALL    B_0
        CALL    B_0
        CALL    B_1         ;停止位 10101
        CALL    B_0
        CALL    B_1
        CALL    B_0
        CALL    B_1
        CALL    TIMER3      ;延时
        CALL    TIMER3
        GOTO    REPEAT

HIDARI  CALL    B_1         ;开始位 1
        CALL    B_1         ;数据 4 位 B1000
        CALL    B_0
        CALL    B_0
        CALL    B_0
        CALL    B_1         ;停止位 10101
        CALL    B_0
        CALL    B_1
        CALL    B_0
        CALL    B_1
        CALL    TIMER3      ;延时
```

```
            CALL        TIMER3
            GOTO        REPEAT
NAKA        CALL        B_1             ;开始位 1
            CALL        B_0             ;数据 4 位 B0000
            CALL        B_0
            CALL        B_0
            CALL        B_0
            CALL        B_1             ;停止位 10101
            CALL        B_0
            CALL        B_1
            CALL        B_0
            CALL        B_1
            CALL        TIMER3          ;延时
            CALL        TIMER3
            GOTO        REPEAT

;发送数据 0 * * * * * * * * * * * * * * * * * * *
B_0         MOVLW       D50             ;12×50 = 600μs
            MOVWF       WWW
            MOVLW       LEDOFF          ;将熄灯数据置入 W 寄存器
            MOVWF       PORTA           ;从端口 A 输出熄灯数据
L0          CALL        TIMER1
            DECFSZ      WWW,1
            GOTO        L0
            RETURN

;发送数据 1 * * * * * * * * * * * * * * * * * * * * *
B_1         MOVLW       D25             ;12×50 = 600μs
            MOVWF       WWW
L1          MOVLW       LEDON           ;将点亮数据置入 W 寄存器
            MOVWF       PORTA           ;从端口 A 输出点亮数据
            CALL        TIMER1
            MOVLW       LEDOFF          ;将熄灯数据置入 W 寄存器
            MOVWF       PORTA           ;从端口 A 输出熄灯数据
            CALL        TIMER1
            DECFSZ      WWW,1
            GOTO        L1
            RETURN
```

```
;时间子程序* * * * * * * * * * * * * * * * * * * * * * * *
TIMER1    MOVLW     D6        ;12μs 定时器子程序
          MOVWF     CNT1
LOOP1     NOP
          NOP
          DECFSZ    CNT1,1
          GOTO      LOOP1
          RETURN

TIMER2    MOVLW     D62       ;0.1ms 定时器子程序
          MOVWF     CNT2
LOOP2     NOP
          DECFSZ    CNT2,1
          GOTO      LOOP2
          RETURN

TIMER3    MOVLW     D100      ;10ms 定时器子程序
          MOVWF     CNT3
LOOP3     NOP
          CALL      TIMER2
          DECFSZ    CNT3,1
          GOTO      LOOP3
          RETURN

          END                 ;程序结束
```

程序清单 6.1　发送器程序

6.2.2 红外线遥控装置的接收器

图 6.12 所示为(与发送器)同样采用 PIC16F84A 的红外线遥控装置接收器的电路图,图 6.13 为实物照片。

接收器采用**3 引脚稳压器** IC 将 9V 电压(1 节 006P 干电池)转换成 5V 稳定电压。LED 的颜色可以根据个人喜好选取。

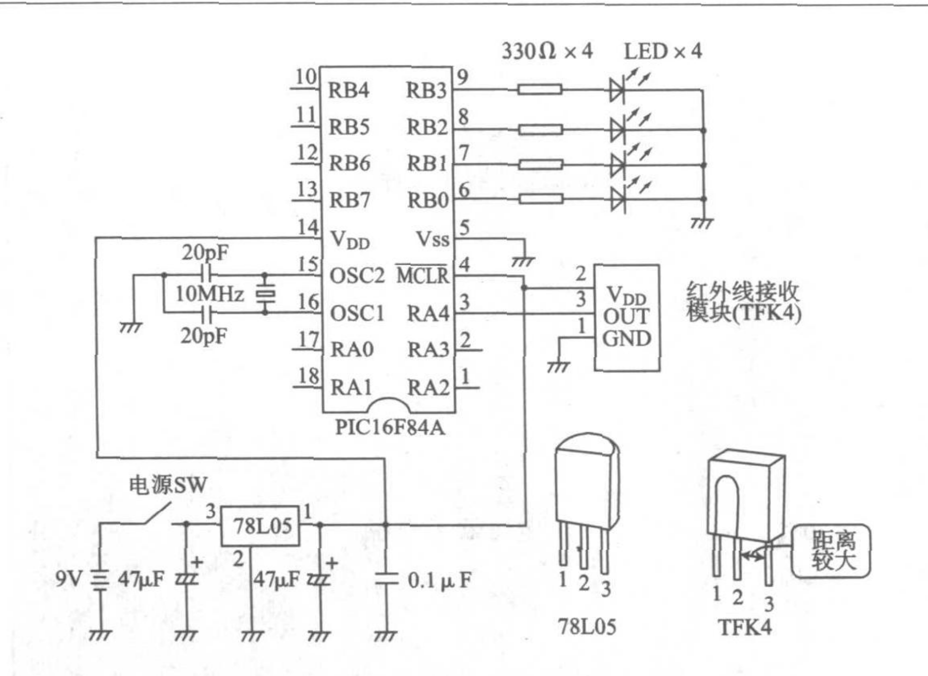


图 6.12　红外线遥控装置接收器电路图

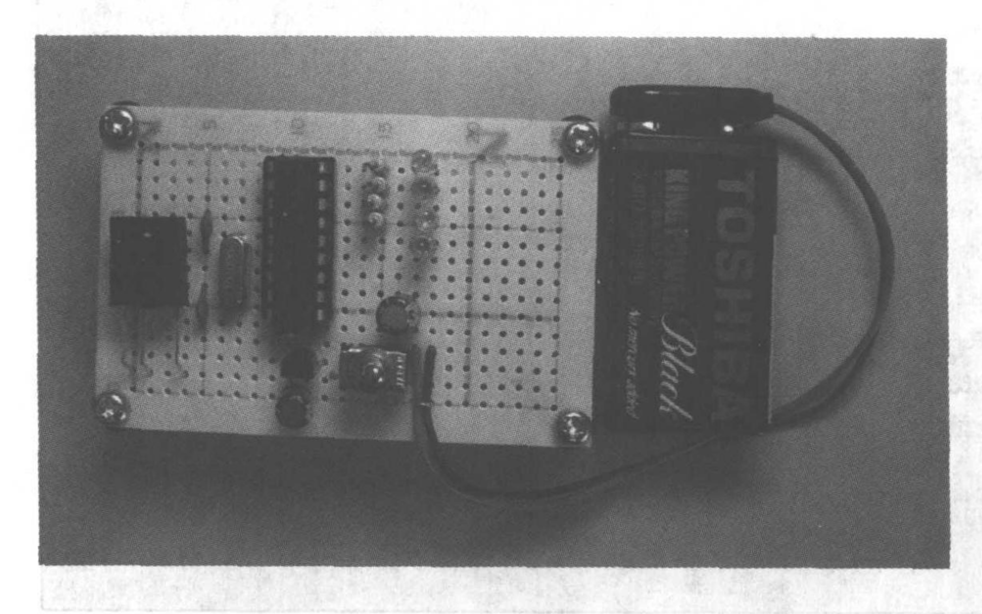


图 6.13　自制红外线遥控装置接收器的实物照片

图 6.14 为接收器的程序框图。

在接收一侧，接收到开始位后大约等待 420μs 再次检查接收模块，如图 6.15 所示。当开始位再次得到确认后，程序就开始以约 600μs 的间隔读取点亮数据位和停止位。如果停止位与“01010”一致，则按照点亮数据的内容点亮（或熄灭）LED；如果二者不一致，则放弃点亮数据，并返回接收待机状态。与发送程序一样，这段时间间隔是由定时器子程序设定的，也包含一定程度的误差 σ。假如传送数据的格式很长，那么越是靠近末位的数据累积误差 σ 越大，这时准确的定时器就显得十分必要了。

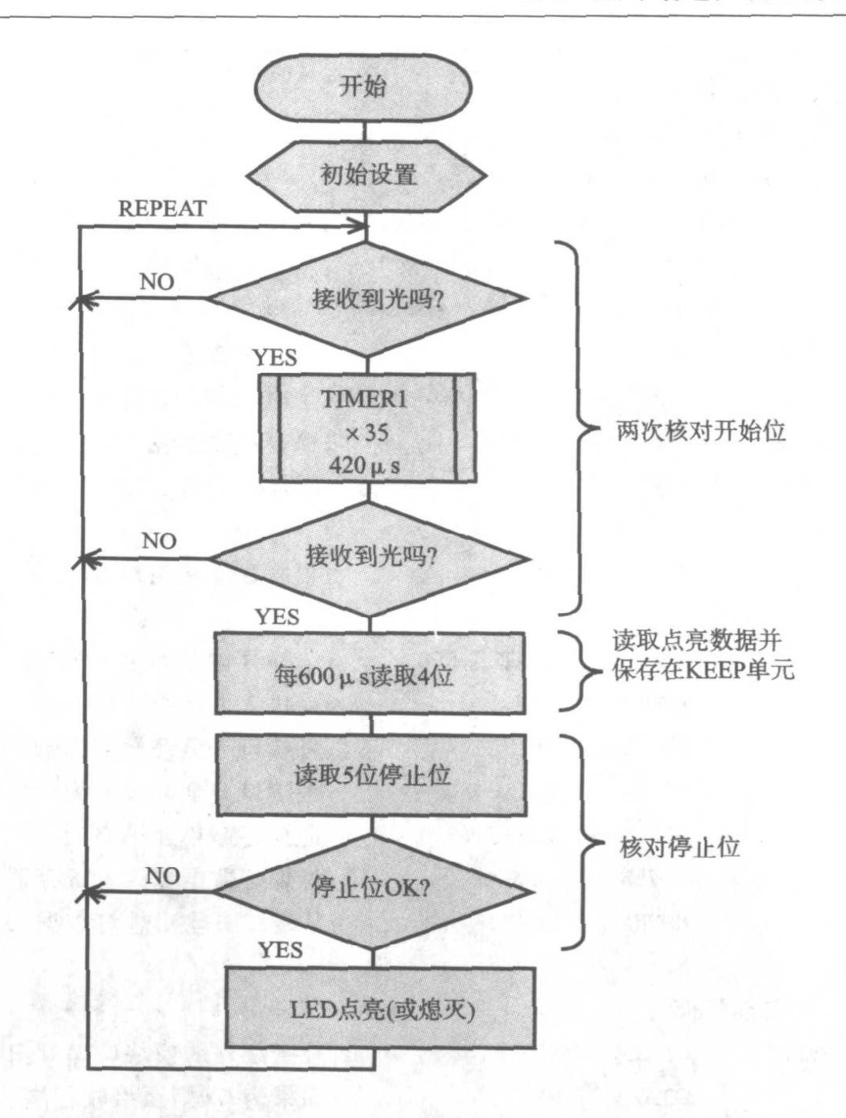


图 6.14 接收器程序框图

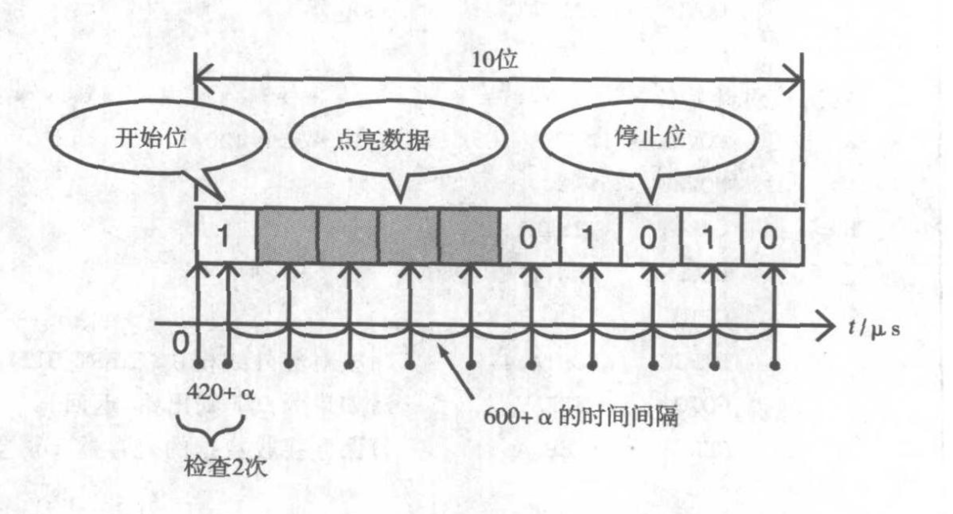


图 6.15 接收模块查验的计时

程序清单 6.2 为接收器的源程序。

```
;程序清单 6.2
;红外接收器源程序
;*****************************
      LIST          P=PIC16F84A      ;指定所使用的 PIC
      INCLUDE       "P16F84A.INC"    ;指定读取设定文件
;***************************
LEDOFF     EQU      00H              ;设定 LED 熄灭数据
CNT1       EQU      0CH              ;定时器的计数变量
KEEP       EQU      0DH              ;保存接收数据
WWW        EQU      0EH              ;载波的计数变量
;**************************
           ORG      0                ;程序存储起始地址

           BSF      STATUS,RP0       ;选择存储单元第 1 列
           MOVLW    1FH              ;端口 A 设定用数据
           MOVWF    TRISA            ;将端口 A 全部设定为输入模式
           CLRF     TRISB            ;将端口 B 全部设定为输出模式
           BCF      STATUS,RP0       ;选择存储单元第 0 列
           MOVLW    LEDOFF           ;将熄灯数据置入 W 寄存器
           MOVWF    PORTB            ;从端口 B 输出熄灯数据

;读取红外模块************************
REPEAT     BTFSS    PORTA,4          ;检查红外线模块(START_BIT)
           GOTO     RX               ;如果为 ON 则至接收程序
           GOTO     REPEAT           ;重复

;两次核对开始位************************
RX         MOVLW    D'35'            ;12×35=420μs
           MOVWF    WWW
CH         CALL     TIMER1
           DECFSZ   WWW,1
           GOTO     CH
           BTFSC    PORTA,4          ;核对红外线模块(START_BIT)
           GOTO     REPEAT           ;如果为 OFF 则出错,返回
           CLRF  ·  KEEP             ;保存接收数据的寄存器 1 清空

;读取发送数据中的 4 位**********************
           MOVLW    D'50'            ;12×50=600μs 动作数据位 0
           MOVWF    WWW
```

```
CL0     CALL      TIMER1
        DECFSZ    WWW,1
        GOTO      L0
        BTFSS     PORTA,4       ;检查红外线模块
        BSF       KEEP,3        ;如果 ON 则设置保存寄存器 1

        MOVLW     D'50'         ;12×50=600μs 动作数据位 1
        MOVWF     WWW
L1      CALL      TIMER1
        DECFSZ    WWW,1
        GOTO      L1
        BTFSS     PORTA,4       ;检查红外线模块
        BSF       KEEP,2        ;如果 ON 则设置保存寄存器 1

        MOVLW     D'50'         ;12×50=600μs 动作数据位 2
        MOVWF     WWW
L2      CALL      TIMER1
        DECFSZ    WWW,1
        GOTO      L2
        BTFSS     PORTA,4       ;检查红外线模块
        BSF       KEEP,1        ;若为 ON 则设置保存寄存器 1

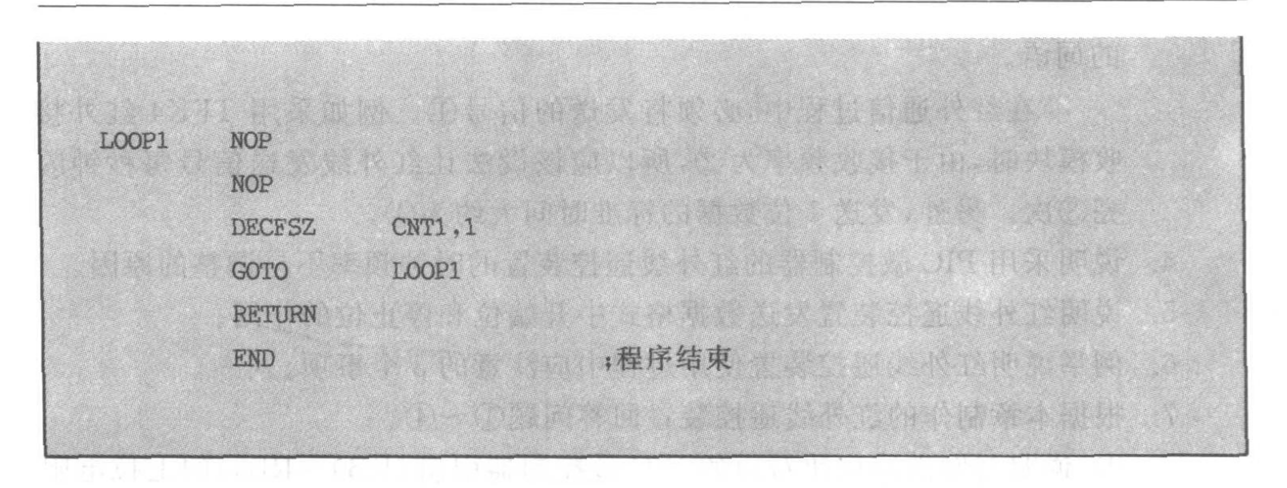
        MOVLW     D'50'         ;12×50=600μs 动作数据位 3
        MOVWF     WWW
L3      CALL      TIMER1
        DECFSZ    WWW,1
        GOTO      L3
        BTFSS     PORTA,4       ;检查红外线模块
        BSF       KEEP,0        ;若为 ON 则设置保存寄存器 1

;核对 5 个停止位(10101)****************************
        MOVLW     D'50'         ;12×50=600μs 动作数据位 4(STOP_BIT0)
        MOVWF     WWW
L4      CALL      TIMER1
        DECFSZ    WWW,1
        GOTO      L4
        BTFSC     PORTA,4       ;检查红外线模块
        GOTO      REPEAT        ;若为 OFF 则出错,返回
```

```
        MOVLW       D´50´         ;12×50=600μs 动作数据位 5(STOP_BIT1)
        MOVWF       WWW
L5      CALL        TIMER1
        DECFSZ      WWW,1
        GOTO        L5
        BTFSS       PORTA,4       ;检查红外线模块
        GOTO        REPEAT        ;若为 ON 则出错,返回
        MOVLW       D´50´         ;12×50=600μs 动作数据位 6(STOP_BIT2)
        MOVWF       WWW
L6      CALL        TIMER1
        DECFSZ      WWW,1
        GOTO        L6
        BTFSC       PORTA,4       ;检查红外线模块
        GOTO        REPEAT        ;若为 OFF 则出错,返回
        MOVLW       D´50´         ;12×50=600μs 动作数据位 6(STOP_BIT3)
        MOVWF       WWW
L7      CALL        TIMER1
        DECFSZ      WWW,1
        GOTO        L7
        BTFSS       PORTA,4       ;检查红外线模块
        GOTO        REPEAT        ;若为 ON 则出错,返回
        MOVLW       D´50´         ;12×50=600μs 动作数据位 6(STOP_BIT4)
        MOVWF       WWW
L8      CALL        TIMER1
        DECFSZ      WWW,1
        GOTO        L8
        BTFSC       PORTA,4       ;检查红外线模块
        GOTO        REPEAT        ;若为 OFF 则出错,返回

;接收数据后的处理* * * * * * * * * * * * * * * * *
        MOVF        KEEP,0        ;将 KEEP1 中的数据转移至 W 寄存器
        MOVWF       PORTA,4       ;OK 则将 W 寄存器的数据从端口 B 输出
        GOTO        REPEAT

;定时器子程序* * * * * * * * * * * * * * * * *
TIMER1  MOVLW       D´61          ;12μs 定时器子程序
        MOVWF       CNT1
```

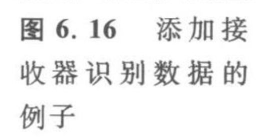


```
LOOP1   NOP
        NOP
        DECFSZ    CNT1,1
        GOTO      LOOP1
        RETURN
        END                ;程序结束
```

程序清单 6.2　发送器程序

发送器和接收器的制作完成后，装好电池并将发送器的红外线 LED 对着接收器接收模块的感光面。这时，一按发送器的按钮，如果对应的 LED 点亮则说明制作成功。中间按钮用于发送 LED 熄灯数据。在进行实际实验的过程中，如果适当调整接收模块红外线的入射角度，那么在数米开外的范围内红外线遥控操作都应该有效。

有时，一台发送器可以同时兼顾几个机器人的动作控制。此时只要往发送数据中添加识别各个接收器的数据即可，如图 6.16 所示。这时，接收器一侧的程序可以预先设置一个条件，即只有接收到自己能够识别的数据之后才能接下去取得动作数据。



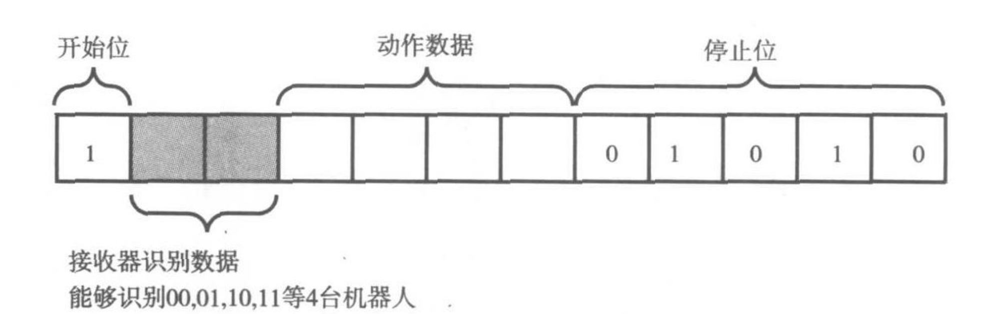


图 6.16　添加接收器识别数据的例子

练习题

1. 回答红外线的波长范围大约为多少纳米？
2. 下面是一则说明红外线 LED 的短文，请在①～④处填入适当的词语。

   “将红外线 LED 通以适当大小的①②就能发光，发出的光线用肉眼③看见，红外线遥控装置④在明亮的室内使用。”
3. 下面是一则说明红外接线收模块使用方法的短文，请在①～④处填入适当

的词语。

“在红外通信过程中，必须将发送的信号①。例如采用 TFK4 红外接收模块时，由于接收频率为②，所以应该设法让红外线发送信号每秒钟闪亮③次。另外，发送 1 位数据的标准时间大约为④。

4. 说明采用 PIC 微控制器的红外线遥控装置的时钟频率不易调整的原因。
5. 说明红外线遥控装置发送数据格式中开始位和停止位的作用。
6. 例举说明红外线遥控装置使用过程中应注意的 3 个事项。
7. 根据本章制作的红外线遥控装置回答问题①～④。

① 说明与发送器按钮对应的 PIC 微控制器引脚(RB0～RB5)的上拉电阻有何作用。

② 说明每次发送完数据后发送器需要设置 10ms 等待时间的理由。

③ 接收器红外线接收模块的输出信号为低电平激活，具体说明理由。

④ 说明通过修改程序提高红外线通信可靠性的途径。

第7章

机器人制作

本书在前面各章分别讲述了微控制器的应用、编程方法、DC 电机的使用方法、红外线遥控装置的功能以及传感器的动作原理等等知识，显然读者可以从中得出一个结论：机器人就是上述诸多技术的集成。

要制造类似在第 1 章中介绍的双足步行机器人那种真正意义上的机器人，对于一般的业余爱好者来说是有相当难度的，其实要理解和掌握机器人基本的动作原理，动手制作不失为一条捷径。

在本章我们就要将迄今学习过的 DC 电机、传感器以及 PIC 组合起来，向制作巡线小车和红外线遥控车发起挑战。

7.1 巡线小车

第 1 章中为了理解微控制器的动作要领，已经简单地认识了巡线小车，本章将具体制作图 7.1 所示的巡线小车。

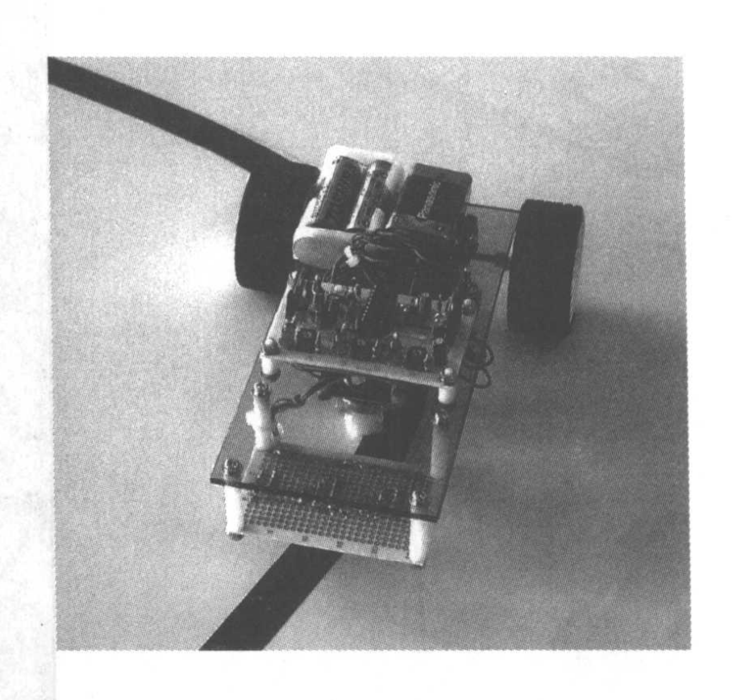


图 7.1　巡线小车照片

7.1.1 巡线小车简介

本章准备实际制作的巡线小车的构思如下：控制采用第 2 章讲解的 PIC 微控制器，在车体前部横向排列 3 个检测引导线条的光传感器，驱动方式在第 1 章也介绍过，即左右后轮各自安装一台驱动电机。巡线小车的外观如图 7.2 所示。

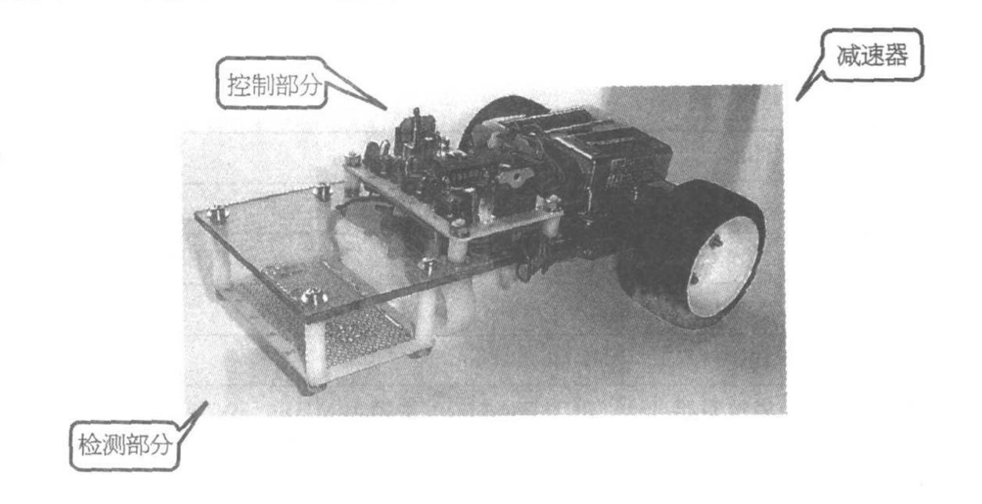


图 7.2 巡线小车的外观

这个巡线小车的动作原理与利用 2 个光传感器的装置基本类似。位于车体前端的光传感器用来检测黑色引导线条，微控制器对检测信号进行处理并控制电机校正偏移量。左右车轮独立驱动，控制小车的方向使之沿引导线自主行走。

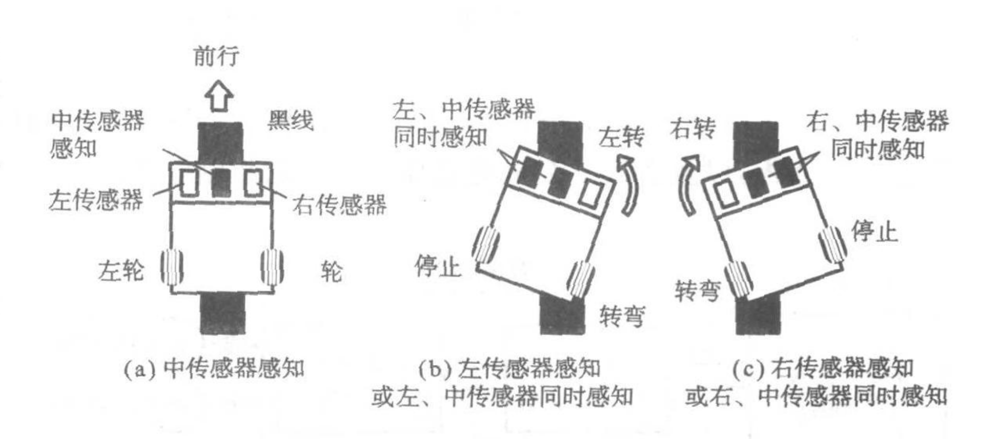


图 7.3 巡线小车的动作原理

以图 7.3 解释巡线小车的动作原理。如果位于中间的传感器(中传感器)检测到黑线，从传感器将发出“有线”信号，后轮的两台电机继续接通运转，结果驱动车体前行。

如果除中传感器之外，左、右传感器中的任一个未检测到黑线，则该传感器输出“无线”信号，这时，脱离引导线的传感器对侧的驱动电机停止运行，同侧的电机继续运行，以此达到校正行进方向的目的，如图 7.3(b)、图 7.3(c)所示。

由此可知，通过包括中传感器在内的 3 个光传感器的配置方式，可以确保检测到引导黑线，实现平缓的控制效果。

下面来讲解如何根据上述动作安排编制控制程序。

在此还需要对引导线条的颜色补充一点，虽然本书以黑色线条为例，但因为可以对光传感器感知的信号做逻辑逆运算，因此稍微改变程序就可以使之用于检测白色线条。

巡线小车的结构参数如表 7.1 所示。

表 7.1　巡线小车的结构参数

本　体	外形尺寸	130mm×190mm×70mm
	底盘	180mm×80mm(丙烯板)
	齿轮箱	带齿轮箱的双联电机(Tamiya 公司 ITEM70097)
	电机	FA-130×2
控制系统	电路板	72mm×47mm(面包板)
	微控制器	PIC16F84A)-20(主频 10MHz)
	电机驱动芯片	TA7257P
检测系统	电路板	72mm×47mm(面包板)
	传感器	光电判读器　东芝　TLP909
电　源	电池	3.0V(5 号电池×2 节)　电机驱动用
		9V(006P 干电池)　电机驱动芯片用
		5V(三引脚稳压器输出)PIC、光电判读器用

巡线小车的结构框图如图 7.4 所示,由两块电路板组成,分别为检测引导线的传感器电路板,以及搭载 PIC 和电机驱动芯片的控制电路板。

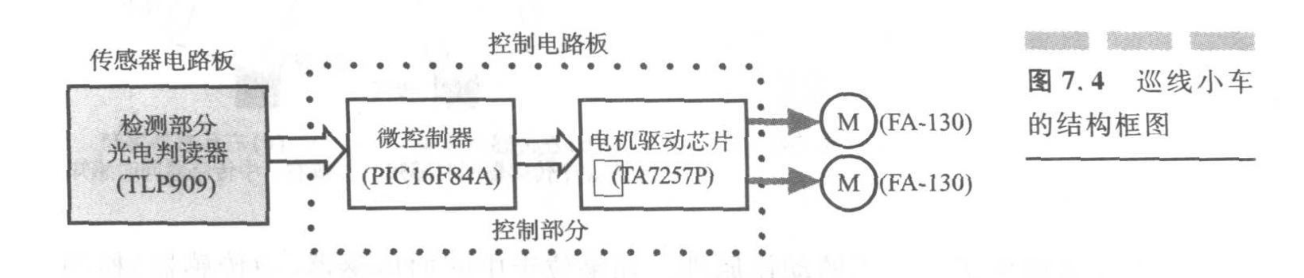


图 7.4　巡线小车的结构框图

传感器检测部分和控制部分分为两块面包板,它们都和齿轮箱共同被固定在**丙烯板**底盘上。电源采用 9V006P 干电池和 2 节 5 号电池,006P 干电池输出端连接 3 引脚稳压器,将电压转换成 5V 输出给 PIC 供电。

控制部分和检测部分也是由 3 引脚稳压器输出的 5V 电压来供电的。电机驱动器芯片需要提供两种电压,分别为 $V_{cc}=9V$,和 $V_s=3V$。然而,由于电机驱动芯片内部存在电压降,测量表明,实际施加到电机上的电压大约为 1.5～2.0V 之间,处于规定的范围之内。

为什么需要如此提供几个规格的供电电压呢?这是因为 PIC16F84A (20MHz)驱动电压的范围为 4.5～5.5V,电机驱动芯片驱动电压的范围 V_{cc} 为 9V, V_s 为 3V,带齿轮箱的双联电机的额定电压为 1.5～3V,既然要分别满足它们就不

得不如此。

需要提醒的是电机运行会产生干扰噪声，因此需要将控制电源和电机驱动电源分开。实践证明，采取这个措施对克服控制系统误动作的效果颇佳。

可见，只要采用了电机驱动芯片就很难避免多种电压供电的麻烦。不过电机驱动芯片确有其诱人之处，如控制电机的正反转，这样即使巡线小车完全脱离引导线条，也可以靠倒车返回到原来的路径。事实上，如果巡线小车具有了倒车功能，那么它的用途就不仅限于巡线了，在加装第 5 章的超声波传感器之后，甚至可以变成自动检测和回避障碍的自主移动机器人。

图 7.5 表示了一个小车的本体，上面安装了 006P 干电池和 2 节电机用 5 号干电池。

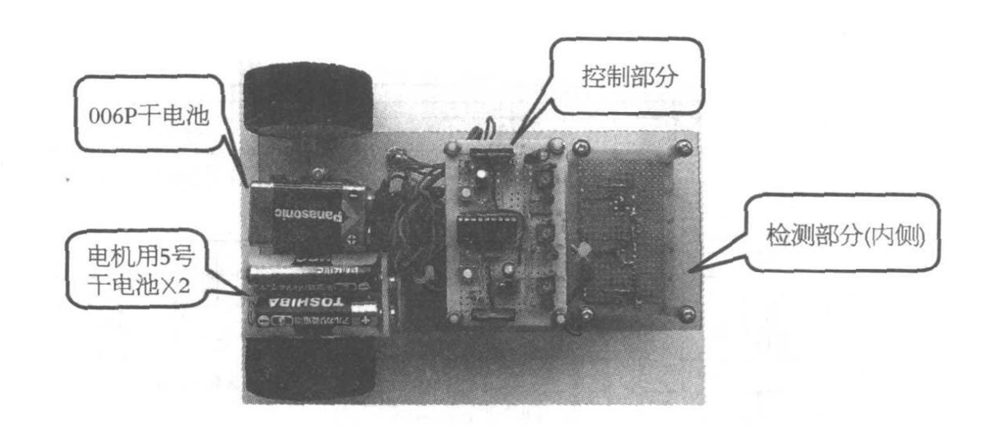


图 7.5 巡线小车的俯视图

从图 7.5 看到，以 PIC16F84A 为核心的控制系统布置在本体的正中，电池置于车轮的上方，干电池盒与齿轮箱减速器连接在一起。将干电池置于车轮上方的用意在于电池的重量直接施加到车轮上，有益于提高车轮的抓地力，使车体运行平稳。

小车本体的腹部如图 7.6 所示，Tamiya 公司生产的带齿轮箱的双联电机安装在本体的后部，驱动车轮旋转，为了减小摩擦阻力，用 1 个塑料材质的万向脚轮与 2 个后轮构成 3 点车体支撑方式。万向脚轮的高度借助垫片调整到使巡线小车的车体保持水平。

光电判读器检测部分的面包板通过垫片固定在底盘上。

选配齿轮箱中齿轮的组合可以改变减速比。减速比大则车轮转速慢，驱动转矩大；反之，减速比小则车轮转速快，驱动转矩小。值得注意是随着行进速度的加快控制的难度会相应地增大，要想很好地完成巡线跟踪任务必须对传感器进行适当的调整，如此一来程序编制的难度也相应提高。

图 7.7 所示为总电路图。表 7.2 为 PIC 端口的分配情况。

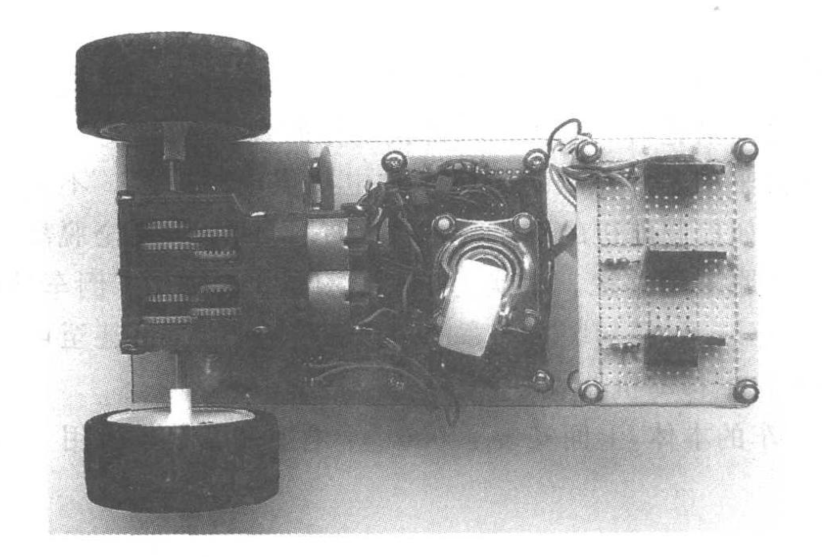


图 7.6　巡线小车的底面

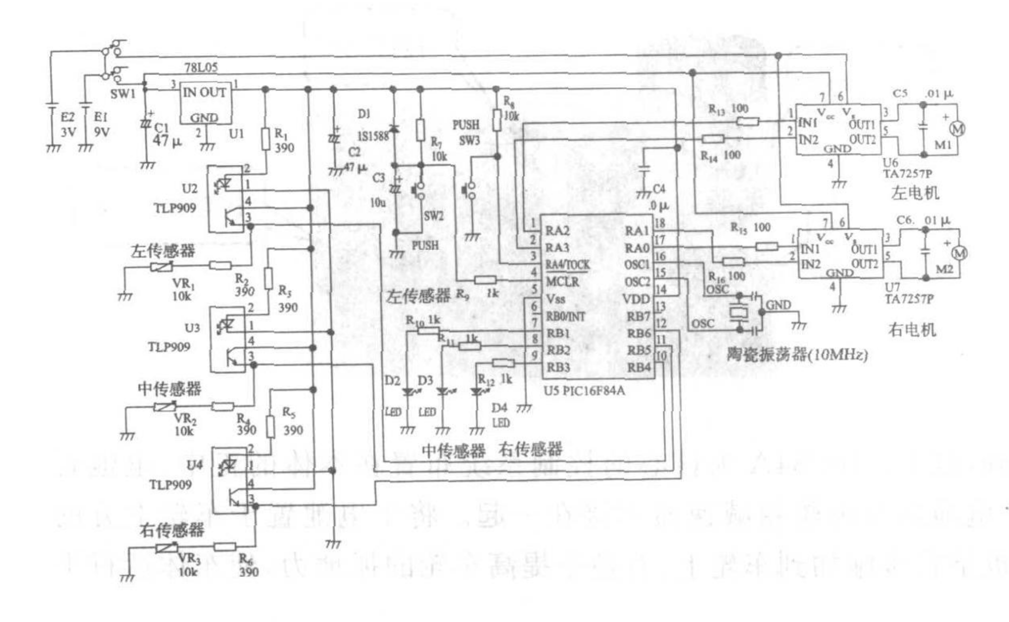


图 7.7　巡线小车电路图

表 7.2　PIC 端口的分配情况

输　入		输　出	
RA4	暂停按钮	RA0	右电机控制信号(IN1)
RB4	左传感器	RA1	右电机控制信号(IN2)
RB5	中传感器	RA2	左电机控制信号(IN1)
RB6	右传感器	RA3	左电机控制信号(IN2)
		RB1	左传感器显示指示灯
		RB2	中间传感器指示灯
		RB3	右传感器指示灯

7.1.2 巡线小车的电路图和零部件

表 7.3 为巡线小车的元器件列表。

表 7.3 巡线小车的元器件列表

符　号	名　称	规格型号	数　量	单价（日元）	备　注
U1	三引脚稳压器	78L05	1	150	
U2～U4	光电判读器	TLP909	3	200	
U5	PIC	PIC16F84A-20	1	350	秋月电子通商
U7,U8	电机驱动芯片	TA7257P	2	330	
D1	二极管	1SS133	1	20	
D2～D4	发光二极管	SLP-881A-37	3	4	秋月电子通商
R_1～R_6	碳膜电阻	390Ω 1/4W	6	5	
R_7,R_8	碳膜电阻	10kΩ 1/4W	2	5	
R_9～R_{12}	碳膜电阻	1kΩ 1/4W	4	5	
R_{13}～R_{16}	碳膜电阻	100Ω 1/4W	4	5	
VR_1～VR_3	可变电阻	10kΩ 1/4W	3	200	
C1,2	电解电容器	47μF　16V	2	30	
C3	电解电容器	10μF　16V	1	25	
C4～C6	陶瓷电容器	0.01μF	3	20	
X1	陶瓷振荡器	10MHz	1	40	秋月电子通商
SW1	小型拨动开关	3P×2	1	160	
SW2,3	触发开关	B3W-1000	2	70	欧姆龙
E1	电池	碱性电池　5号	2	150	
E2	电池	006P　9V	1	500	
	电池盒	5号×2	1	150	
	电池盒		1	30	
	IC 插座	18PIN	1	50	秋月电子通商
	面包板	71×47mmC 型	2	70	秋月电子通商
	带齿轮箱双联电机	ITEM70097	1	700	Tamiya
	车轮	ITEM70111	1	450	Tamiya
	丙烯板（透明）	180mm×80mm×3mm	1	200	
	万向脚轮	小型塑料	1	100	
	垫片	BR-325A 25mm×Φ3	4	65	广杉仪器
	垫片	BR-308A 8mm×Φ3	4	55	广杉仪器
	垫片	BR-306A 6mm×Φ3	4	50	广杉仪器
		螺钉	3M	10	
	弹簧垫圈	3M	10		
	垫圈	3M	10		
	导线		适量		
合计金额（估计值）			约 6000 日元		

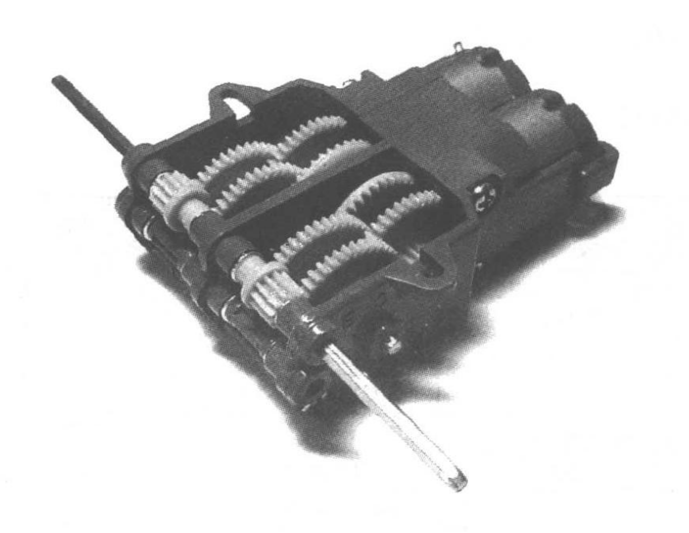


图7.8 带齿轮箱的双联电机(摘自Tamiya株式会社,减速比58:1)

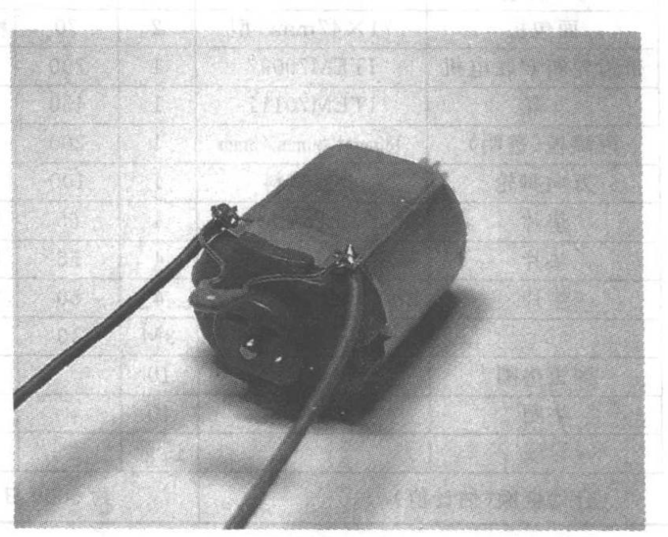
下面逐一说明巡线小车的各大部件。

该小车采用图7.8所示Tamiya公司生产的带齿轮箱的双联电机,它内装2个FA-130电机,通过改变齿轮的组合可以按表7.4选择减速比。

表7.4 减速比和转矩

减速比	58.2:1	203.7:1
转矩(g·cm)	419	1404
转速(r/min)	227	65

在将电机嵌入齿轮箱之前应预先接好导线和防干扰用的电容器,电容器应该尽量靠近电机的接线端子,如图7.9所示,但不要影响电机安装。



(已接入电容和导线的状态)

图7.9 FA-130电机

由表 7.4 可知，齿轮箱的减速比有两种选择，如果希望减小巡线小车车体动作控制的难度，可以采用低速运动，将齿轮箱的减速比设为低速档 203.7：1，选择减速比时应考虑到电池电压的匹配问题。

本例的齿轮箱按照公司产品说明图例中的 C 型（低速状态）组装，两个齿轮箱呈双联组装的形式。实际上组装的形式有 A、B、C 三种，各个转轴的安装高度不同，所以将其安装到底盘上之前应预先选择好适当的型号。

带齿轮箱的双联电机的安装尺寸如图 7.10 所示，加工底盘之前可以此为参考。

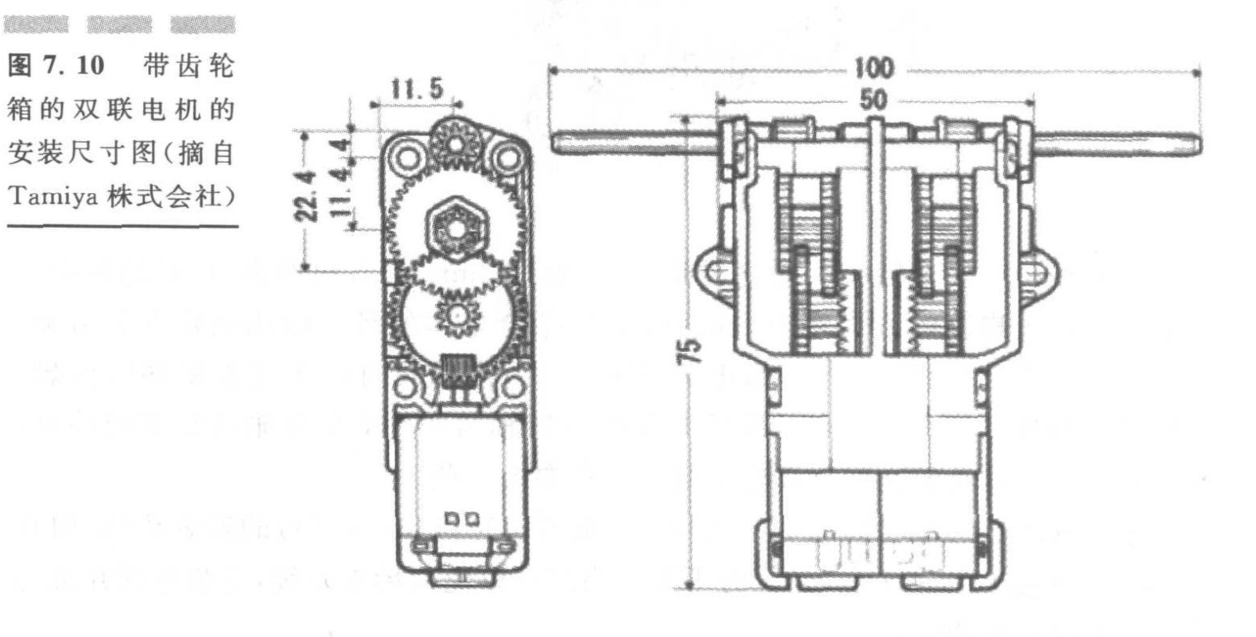


图 7.10 带齿轮箱的双联电机的安装尺寸图（摘自 Tamiya 株式会社）

组合车轮和万向脚轮的外观分别如图 7.11、图 7.12 所示，车胎采用 Tamiya 生产的玩具赛车车轮组件。

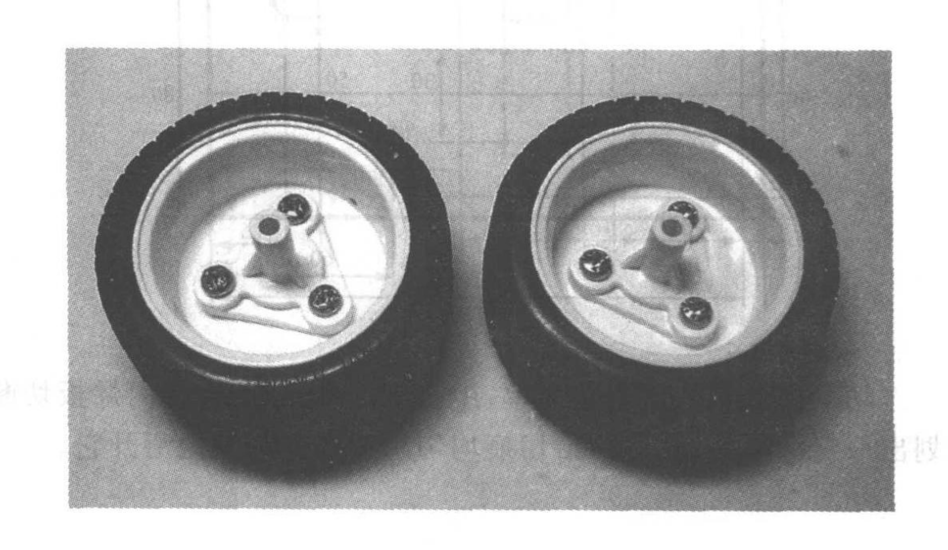


图 7.11 玩具赛车车轮组件图

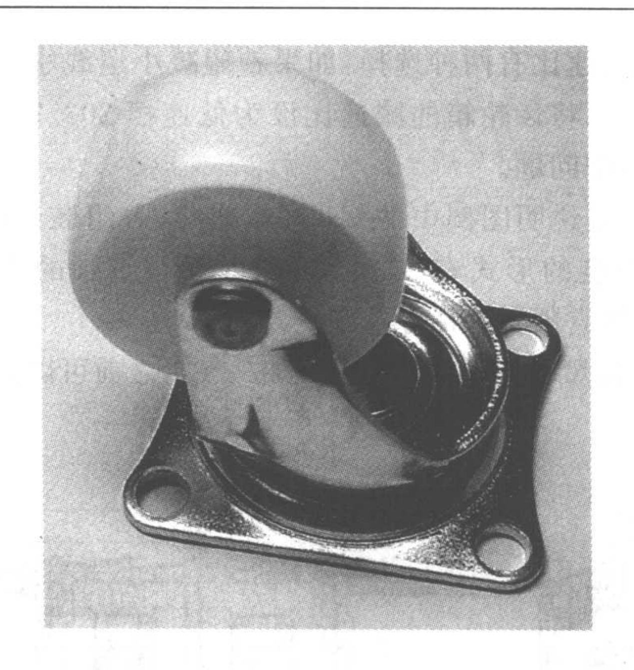


图 7.12　万向脚轮

该车轮由一套转轴和轮毂组装而成，转轴是 3mm 的六角形轴，转轴的间距可以利用转轴上螺纹的方向调节，在组装时应结合本体的具体结构决定安装方向。万向脚轮可从五金商店购买，高度为 35mm。万向脚轮有两个任意旋转的转轴。安装时需要略加注意的一点是脚轮车轮的自转轴与脚轮立轴的轴线要彼此错开，因此脚轮回转运动的空间比脚轮本身的半径要大一些。

底盘丙烯板的尺寸如图 7.13 所示，上面开有穿过 3mm 螺钉的安装孔（使用直径 φ3.2mm 左右的钻头）。钻孔时不要用力过猛以免丙烯板开裂，安装电源开关的孔应与螺纹尺寸相匹配。

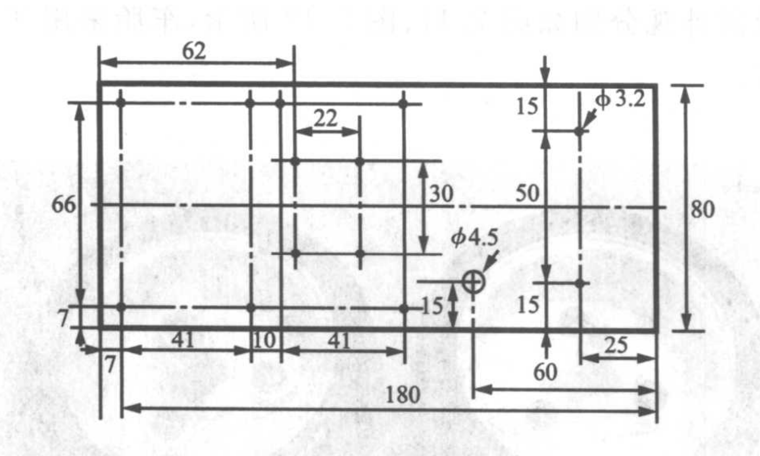


图 7.13　底　盘

切割丙烯板的操作很简单，用图 7.14 所示的专用丙烯板切断刀在板的两侧各划出一定深度的切槽后，沿切痕均匀用力就能轻松地掰开它。

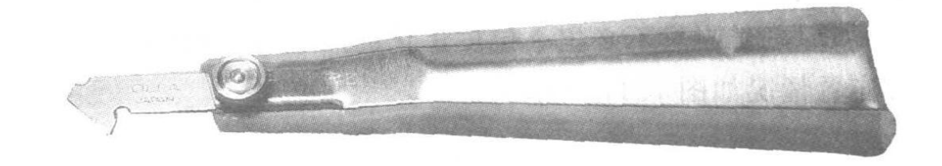


图 7.14 丙烯板专用切断刀

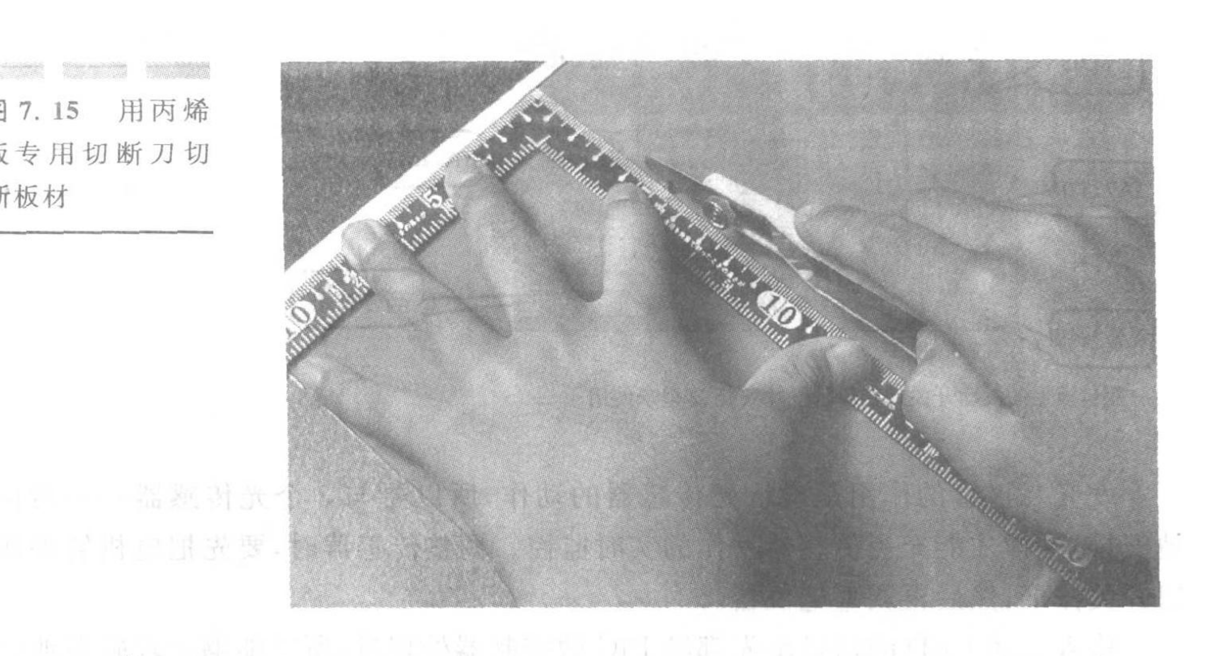


图 7.15 用丙烯板专用切断刀切断板材

丙烯板专用切断刀的用法不同于一般的小刀，如图 7.15 所示要用手拉动刀柄。

检测部分如图 7.16 所示，应按照引导线条的宽度布置 3 个传感器的间距，并用锡焊将传感器固定牢靠。面包板上电阻的作用是光电判读器内部 LED 的限流。检测部分的元件一般无须做特别的调整，因此可以将连线弯成直角并焊牢。需要特别注意的是 3 个光电判读器检测孔的高度应该尽量保持一致。

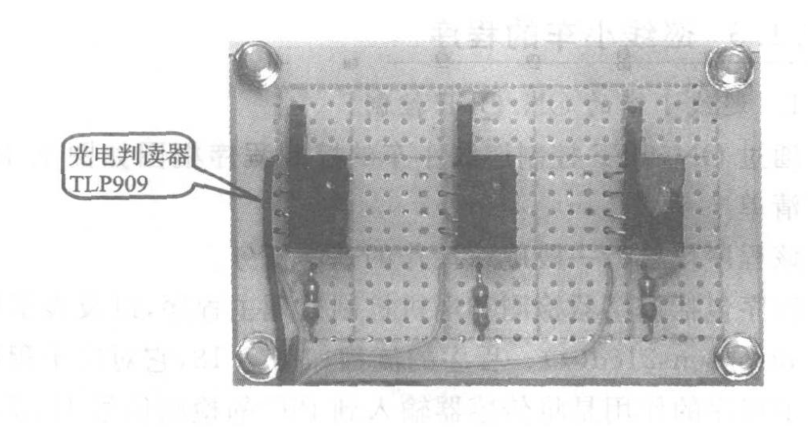


图 7.16 检测部分

提高布线的效率还是需要一些技巧的。在面包板的底面直接利用元器件本身

的引出线连接即可，位于面包板正面的元器件则可以用塑包导线连接。如果是耐热导线，**焊接**时当然不必担心包皮焦化的问题，因而操作十分方便。控制部分的元器件布置情况如图 7.17 所示。

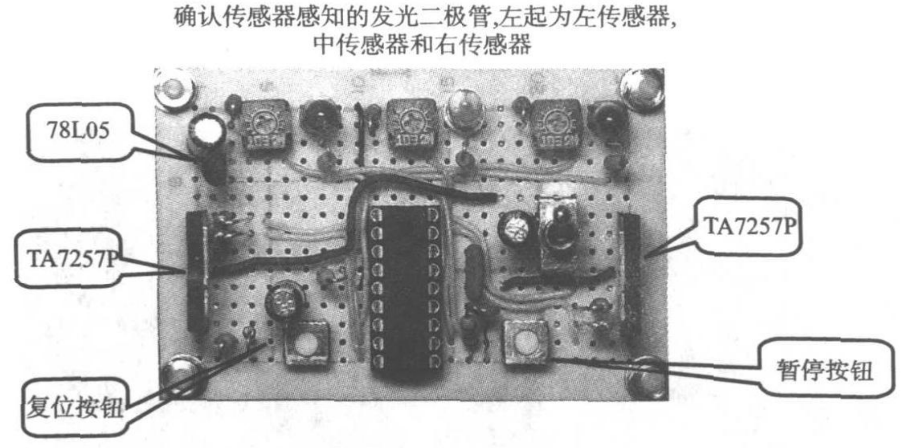


图 7.17　控制部分外观图

发光二极管的作用是确认光传感器的动作，所以要与 3 个光传感器一一对应地安排，这样才能充当传感器动作的实时监控。调整传感器时，要先把电机暂停按钮开关按下，然后再调整电位器。

输入上述 LED 的信号事先都经 PIC 微控制器处理过，所以能够比较确实地反映传感器的反射状态。如果引导线条是黑色的，只要适当地调整电位器使传感器感知到黑色条带时 LED 点亮，而感知到周围的白色部分时 LED 熄灭。这样的调整应该反复进行以达到传感器的最佳工作状态。

电子元器件的耐热性都比较差，因此在焊接时烙铁头接触管脚的时间不宜过长。同时也应注意不要烫坏面包板。

7.1.3　巡线小车的程序

1. 巡线小车的 ON/OFF 控制

通过 ON/OFF 控制巡线小车的**控制程序**框图如图 7.18 所示，程序的列表如程序清单 7.1。

该程序仅执行一段确认动作的基本任务。

程序包括传感器检测和电机控制处理主程序，以及表示检测结果的子程序(函数名 disp_sens2led())。程序的流程如图 7.18，它对应于程序清单 7.1。

子程序的作用是将传感器输入到 PIC 的检测信号用 LED 指示出来，并用～将对应的位取逆，代入 RB1<＝RB4(左传感器)。当传感器感知到黑线时，向 PIC 的 RB4 输入“0”。

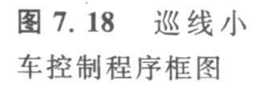
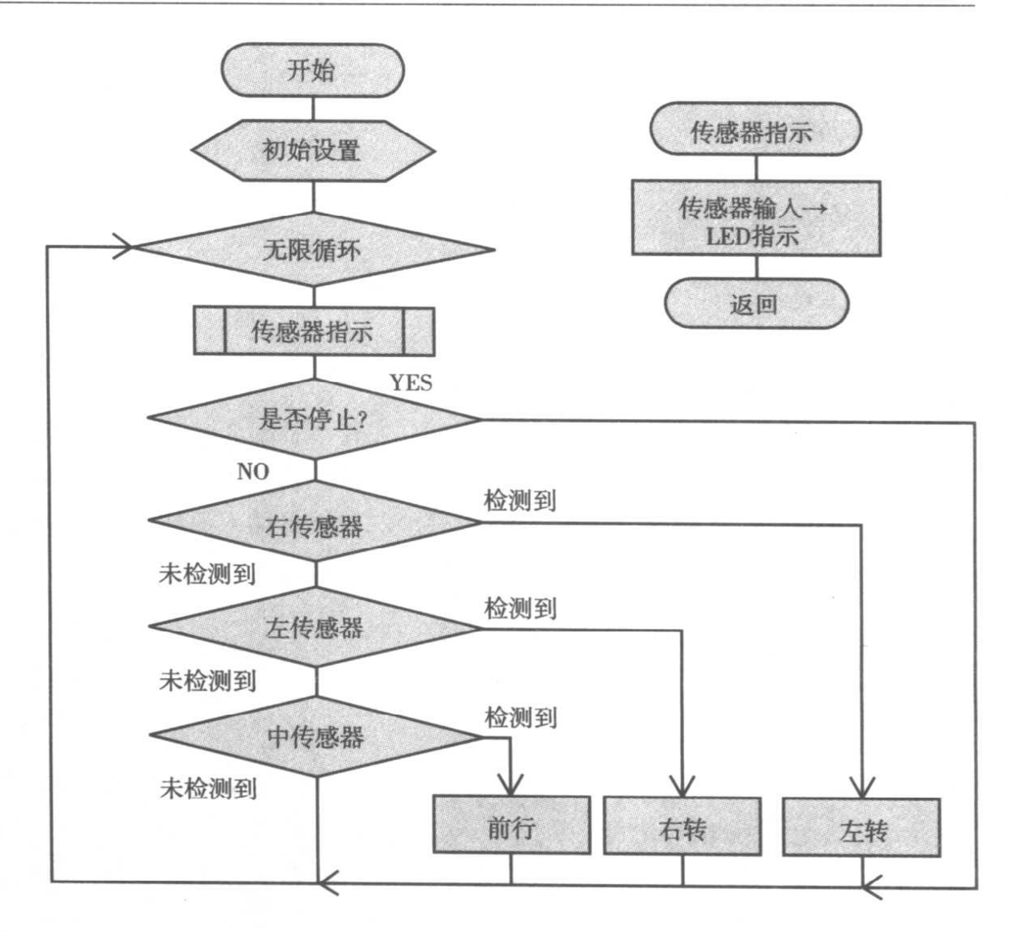


图 7.18 巡线小车控制程序框图



主程序利用 while 语句进行无限循环实时读取传感器的值，并以此为条件通过 if 语句进行判断，将相应的电机控制信号输出到 PORTA，最终控制电机的驱动芯片 TA7257P。

上述处理程序是一段最基本的程序，仅仅通过判断 3 个传感器中任意一个的状态来控制电机的 ON/OFF，所以车体行进间会表现出不停的摇摆。

根据上述程序的安排，传感器状态的判别需要逐个地进行，所以如果线条宽度过窄（甚至窄到小于相邻两个传感器的间距），或者小车彻底偏离引导线条，它就只好无奈地停止下来。因此可知程序需要经过改进才能实现更完善的功能和高难度的运行。

图 7.19 表示一台巡线小车在行走测试时的轨迹。

```
// 程序清单 7.1
// 巡线小车的控制程序
#include <pic.h>                    // 头文件
__CONFIG(0xFFF2);                   // 配置位

// 传感器输入→LED 指示
```

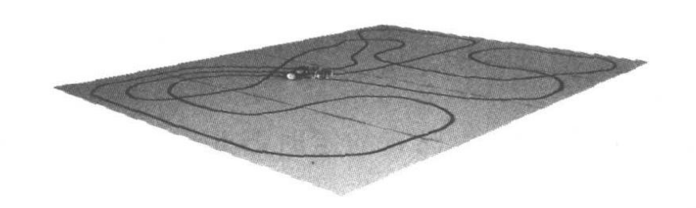
```
disp_sens21ed()
{
    unsigned char a,b,c;

    a = RB4;
    b = RB5;
    c = RB6;
    RB1 = ~a;                    // 左传感器输入反转→左 LED 指示
    RB2 = ~b;                    // 中传感器输入反转→中间 LED 指示
    RB3 = ~c;                    // 右传感器输入反转→右 LED 指示
}

// 主程序
main()
{
    TRISA = 0xF0;                // 设置端口 A 上位 1 位输入和下位 4 位输出
    PORTA = 0x00;                // 端口 A 清零
    TRISB = 0x71;                // 设置端口 B 上位 1 位输入和下位 4 位输出
    PORTB = 0x00;                // 端口 B 清零

    While(){                     // 反复循环
        disp_sens21ed();         // 将传感器数据输出到 LED
        if(RA4 = = 1){           // 停止否?
        if(RB6 = = 0){           // 如果右传感器处在黑线上则右转
        PORTA = 0x08;            // 右转
        }else if(RB4 = = 0){     // 如果左传感器处在黑线上则左转
        PORTA = 0x01;            // 左转
        }else if(RB5 = = 0){     // 如果中传感器处在黑线上则直行
        PORTA = 0x09;            // 前行
            }
        }else PORTA = 0x00;      // 调整传感器时停止
        }
    }
```

程序清单 7.1　巡小车的控制程序


图 7.19 巡线小车行走测试的轨迹

2. 巡线小车的 PWM 控制

为了改善程序清单 7.1 的性能，可以采用 PWM 进行车轮运转速度控制。当两个传感器感知到引导线条，点亮指示灯并准备做出改变车体行进方向的响应时，靠程序的 PWM 控制降低电机转速，实现平滑的转向过程。在图 7.20 中表示出一个电机停止旋转，另一个电机在 PWM 的控制之下。

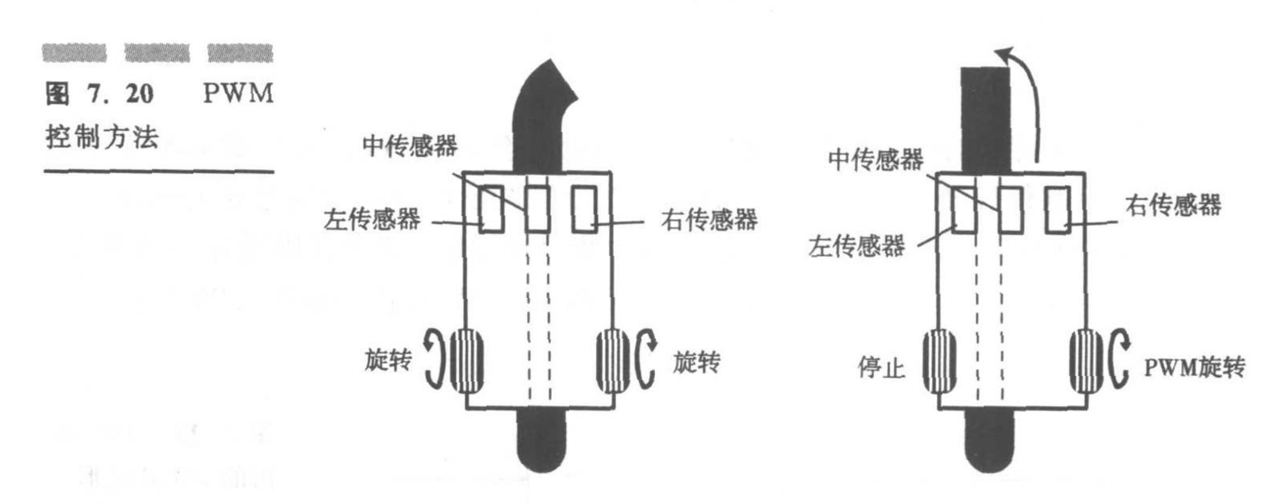


图 7.20 PWM 控制方法

有关 PWM 控制的基本原理已经在第 3 章“直流电机的控制”中做过介绍。基本的思想就是通过反复循环控制改变 ON/OFF 的时间分配。但巡线小车无法借助循环处理实现 PWM，需要通过图 7.21 所示的中断处理方式实现。

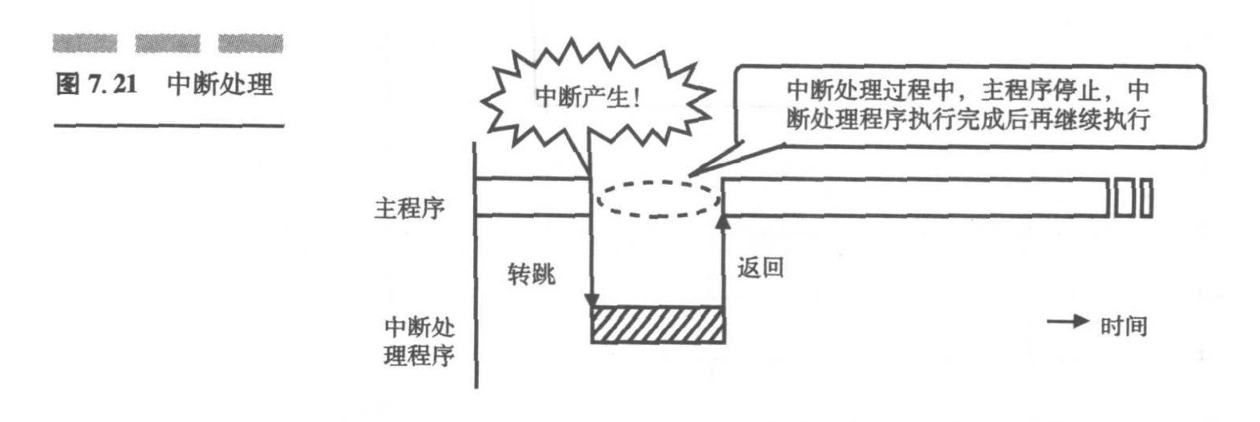


图 7.21 中断处理

如图 7.21 所示，所谓中断就是执行中的程序暂时停止一定时间，以便处理优先级更高的其它程序的一种程序结构。中断处理程序完成后，立即重新处理前面

被打断的程序。

通过中断产生 PWM 信号的具体方法如图 7.22 所示。

首先,需要对 PIC 中已有的**定时器 0** 设定中断周期。每次中断发生时都转跳到名为 intPWM()的函数,变量 cnt 在内部对中断次数计数。

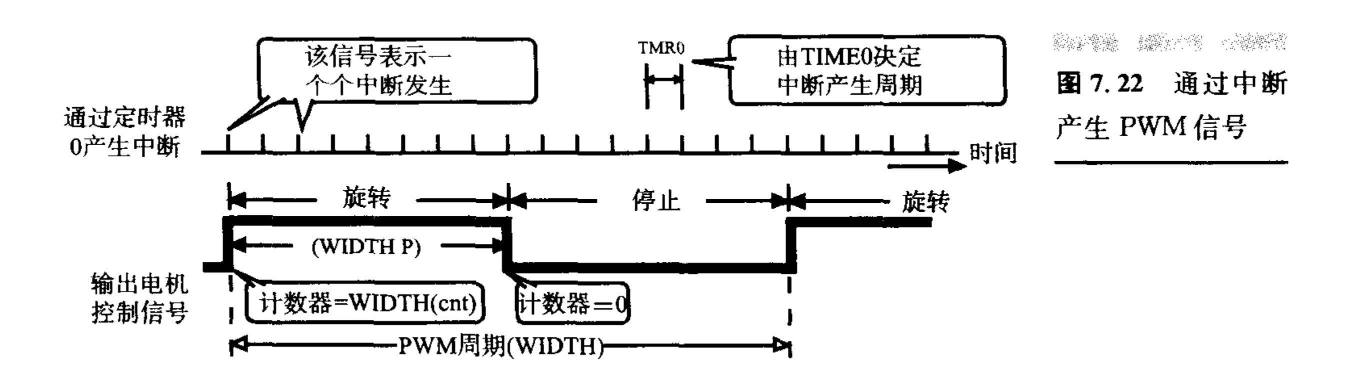


图 7.22　通过中断产生 PWM 信号

WIDTH 定义的次数需要预先在 cnt 中设定,然后做减法,凡计数器不为 0 均输出电机控制信号 dutySET=1,当计数器为 0 则输出电机控制信号 dutySET=0。

电机的 PWM 控制依据由上述信号与主程序中 while 语句无限循环内的 PORTA 两者的逻辑"乘"(&)的结果进行的。实际测得的 PWM 信号如图 7.23 所示。

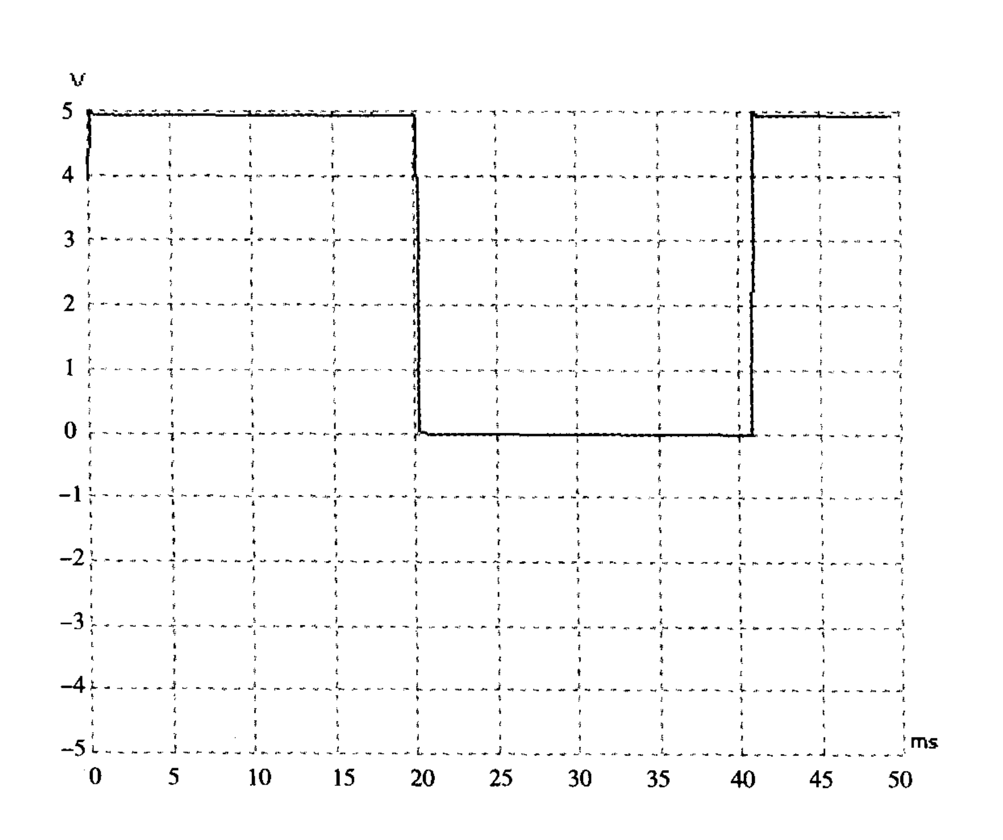
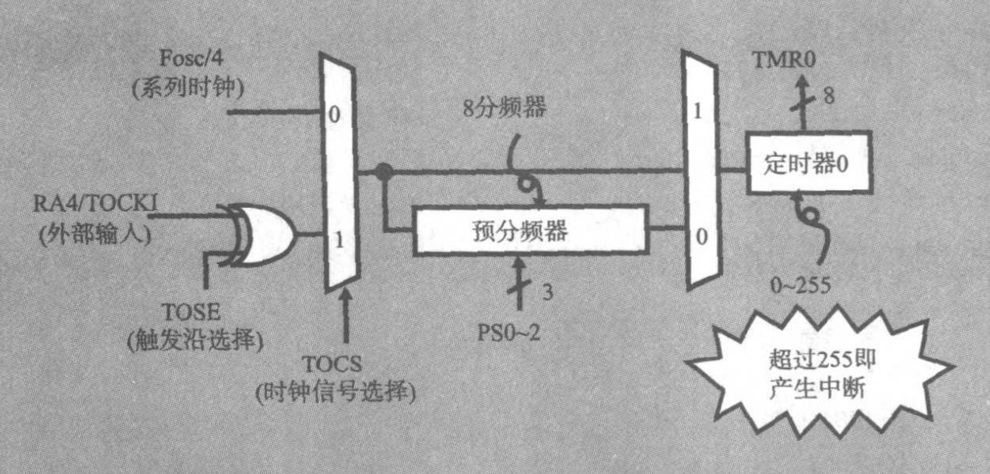


图 7.23　PIC 输出的 PWM 波形

专栏：何调定时器 0

PWM 控制所使用的 PIC 定时器 0 的结构如下图。



利用计时器 0 可以得到任意时间的定时器，计算时间的关系式为

(时钟频率÷4)×预分频器值÷计数数值

例如，假设时钟频率为 10MHz，定时器设定为最大，则定时器的时间为

(10MHz÷4)×(1/256)÷255=38.147Hz≈26.2ms

如果应用 PICC LITE，则可通过"OPTION=设定值;"进行设定。

(例如)OPTION=0xD0;(预分频器值 1/2)

预分频器的值由最后的 4 位即定时器的 PS0～2 决定，可以在 000(1/2)/～111(1/256)之间任意设定。计数过程在 0～255 之间反复循环，TMR0 达到 255 后即返回到 0。

从图 7.23 可知，周期约为 40ms，ON 的脉冲宽度为 50%。在程序清单 7.2 中通过下面的命令设置这些值。

```
#define WIDTH        200        // 脉冲周期
#define WIDTHP       200        // PWM 脉冲 ON 的宽度
#define TCOUNT       100        // 定时器的计数值
```

利用 WIDTH 和 WIDTHP 可以决定 ON/OFF 的比例。需要提醒的是中断产生的时间间隔也可以通过振荡频率和 OPTION 的设定值来改变。

请注意，PWM 的频率即使达到数十千赫也都能够满足平滑控制，但是定时器 0 的中断响应却不怎么快。而且电机也存在一个变化速率匹配的问题，所以不妨通过不断的尝试，适当地改变上述设定值以便得到最佳的效果。

```
// 程序清单 7.2
// 巡线小车  PWM 控制程序
#include"pic.h"                          // 头文件
-CONFIG(0xFFF);                          // 配置位
#define WIDTH       200                  // 脉冲周期
#define WIDTHP       100                 // PWM 脉冲 ON 的宽度
unsigned int cnt = WIDTH;                // 周期变量
unsigned char dutySET = 0xFF;            // duty 设定用变量
disp_sens21ed()                          // 传感器输入→LED 指示
{
        unsigned char a,b,c;

        a = RB4;
        b = RB5;
        c = RB6;
        RB1 = ~a;                        // 左传感器输入反转→左 LED 指示
        RB2 = ~b;                        // 中传感器输入反转→中间 LED 指示
        RB3 = ~c;                        // 右传感器输入反转→右 LED 指示
}

main()                                   // 主程序
{
        OPTION = 0xD0;                   // 定时器 0 时钟,为内部时钟、无分频器
        TMR0 = 0;                        // 将 0 写入定时器 0
        T0IF = 0;                        // 将定时器 0 的中断标志设为 0
        T0IE = 1;                        // 使用定时器 0 中断
        GIE = 1;                         // 打开中断服务
        TRISA = 0xF0;                    // 设置端口 A 上位 1 位输入,下位 4 位
                                         // 输出
        PORTA = 0x00;                    // 端口 A 清零
        TRISB = 0x71;                    // 设置端口 B 上位 1 位输入,下位 4 位
                                         // 输出
        PORTB = 0x00;                    // 端口 B 清零

        While(1){                        // 反复循环
            disp_sens21ed();             // 将传感器数据输出到 LED
```

```
            if(RA4 = = 1){                          // 停止否?
              if(RB4 = = 0&&RB5 = = 0){
                                                    // 如果中、左传感器处在黑线上则
                                                    // 左转
              PORTA = 0x01&dutySET;                 // PWM 左转
        }else if(RB5 = = 0&&RB6 = = 0){
                                                    // 如果中、右传感器处在黑线上则
                                                    // 右转
              PORTA = 0x08&dutySET;                 // PWM 右转
            }else if(RB4 = = 0){                    // 如果左传感器处在黑线上则左转
              PORTA = 0x01;                         // 左转
            }else if(RB6 = = 0){                    // 如果右传感器处在黑线上则右转
              PORTA = 0x08;                         // 右转
              }else if(RB5 = = 0){                  // 如果中传感器处在黑线上则直行
              PORTA = 0x09;                         // 前行
            }
            }else PORTA = 0x00;                     // 调整传感器时停止
            }
}

void interrupt intPWM()                             // 中断函数
{
        if(T0IF){                                   // 如果是定时器 0 则中断
            T0IF = 0;                               // 将计时器 0 的中断标志设为 0
            if(cnt< = WIDTHP)     dutySET = 0xFF;
                                                    // 将 dutySET 的内容设为 0xFF
            if(--cnt = = 0){                        // 判断 cnt-1 的结果是否为 0?
                cnt = WIDTH;                        // 将周期写入 cnt
                dutySET = 0x00;                     // 将 dutySET 的内容设为 0
            }
        }
}
```

程序清单 7.2　巡线小车 PWM 控制程序

程序清单 7.2 的结构框图如图 7.24 所示。

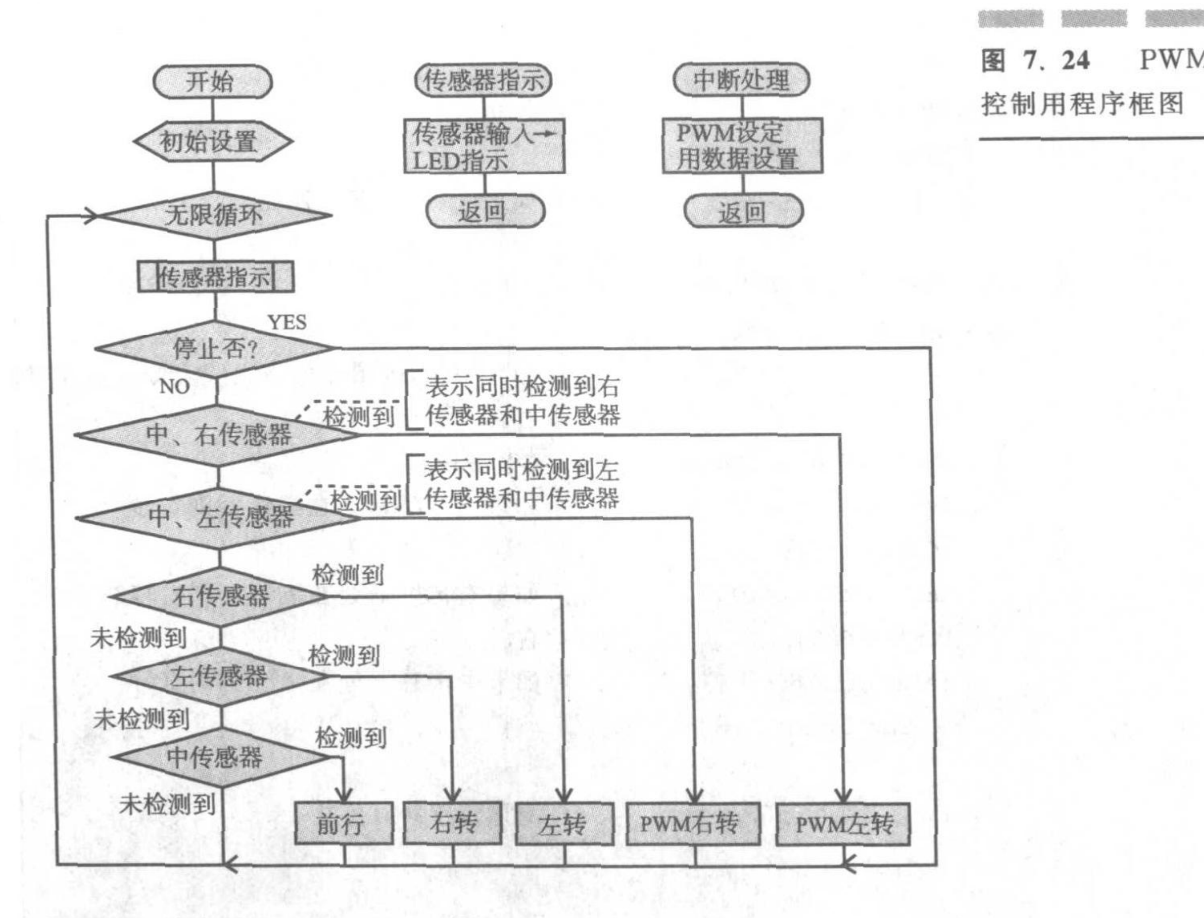


图 7.24　PWM 控制用程序框图

7.2 遥控坦克

巡线小车是根据自身判断跟踪黑色线条的自主机器人。下面再来制作一种接收操作员的指令，并**按照指令动作的机器人**。

本节打算利用第 6 章制作的**红外线遥控装置**，制作一辆可以前进、后退、右转、左转和停止的**机器人坦克**（参见图 7.25）。

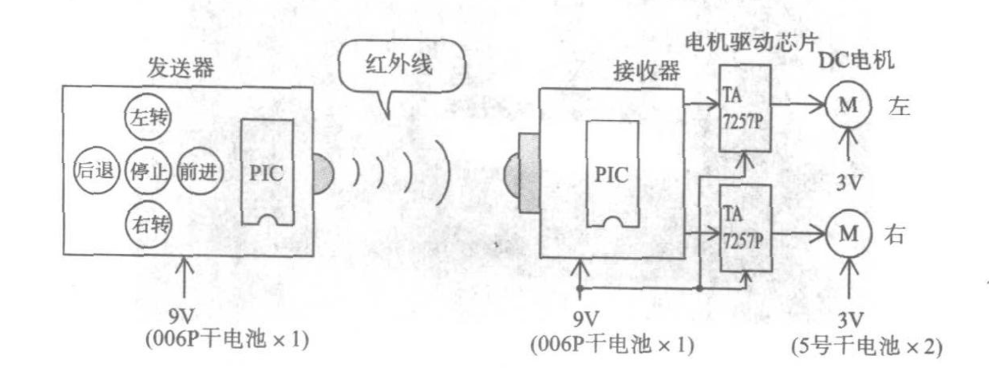


图 7.25　制作红外线遥控坦克

7.2.1 遥控坦克概述

图 7.26 为遥控坦克的构成概念图。图 7.27 所示为完成的自制遥控坦克红外线发送器(左)和车体(右)的照片。

图 7.26 遥控坦克的构成



红外线发送器采用与第 6 章相同的电路,仅仅需要改变一下程序中的发送数据而已。接收器也与第 6 章制作的基本相同,只是用 2 个 DC 电机驱动芯片取代原来的 4 个 LED,用以控制左右的小型 DC 电机。

这里,坦克的车体从田宫模型趣味制作系列中挑选**推土机机械基本模块**(参见图 7.28,型号:ITEM70104,定价:2200 日元)的机座部分借用。当然读者们也不妨发挥各自的想像,改成全部自制。

图 7.27 遥控坦克的红外线发送器(左)和车体(右)照片

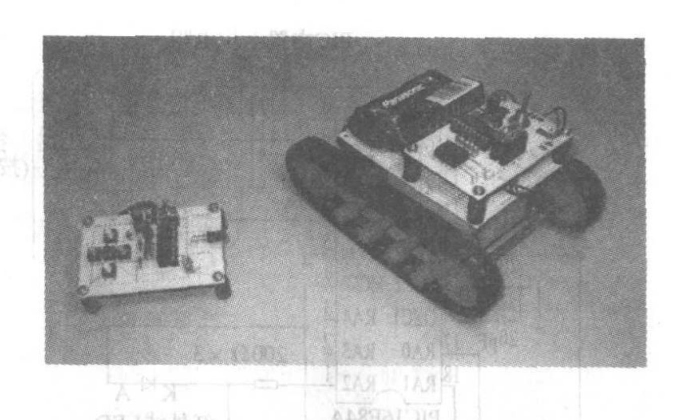
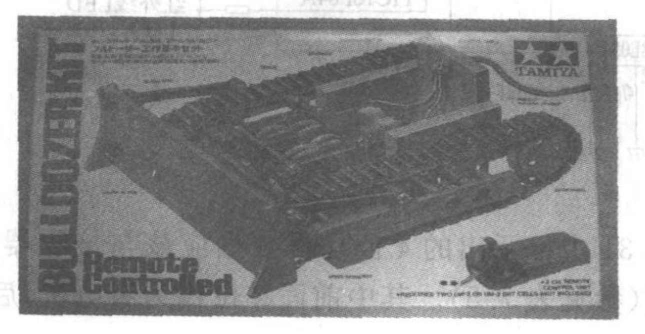


图 7.28 推土机机械基本模块(仅取机座部分)



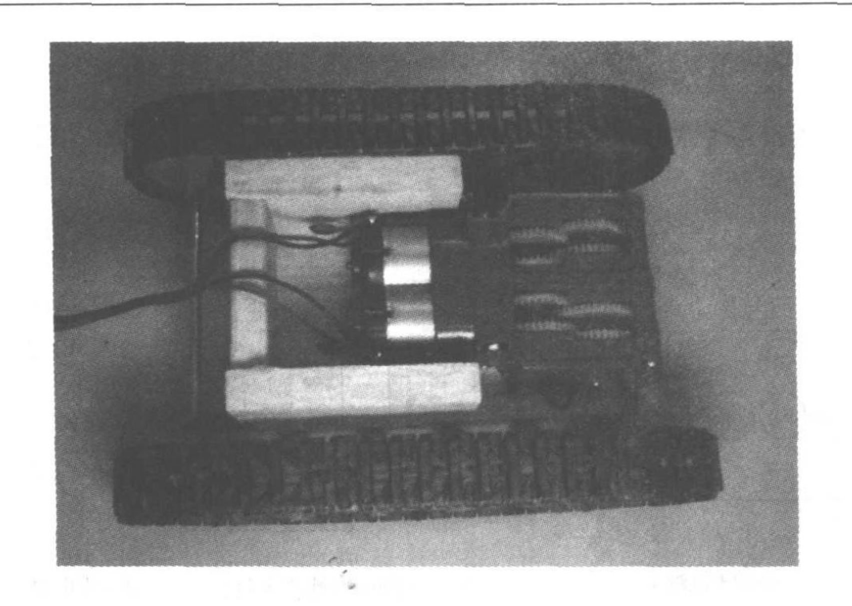


图 7.29　选用的机座部分

7.2.2 遥控坦克的发送器

图 7.30 为遥控坦克发送器的电路图。该电路与第 6 章图 6.9 相同。在图 6.9 中，发送器的按钮与接收器的 LED 相对应，在本节则改为分别指挥遥控坦克前进、后退、右转、左转和停止的开关。

发送的动作数据应该按图 7.31 所示的格式进行变更，不过开始位（"1"）和停止位（"01010"）无需改动。

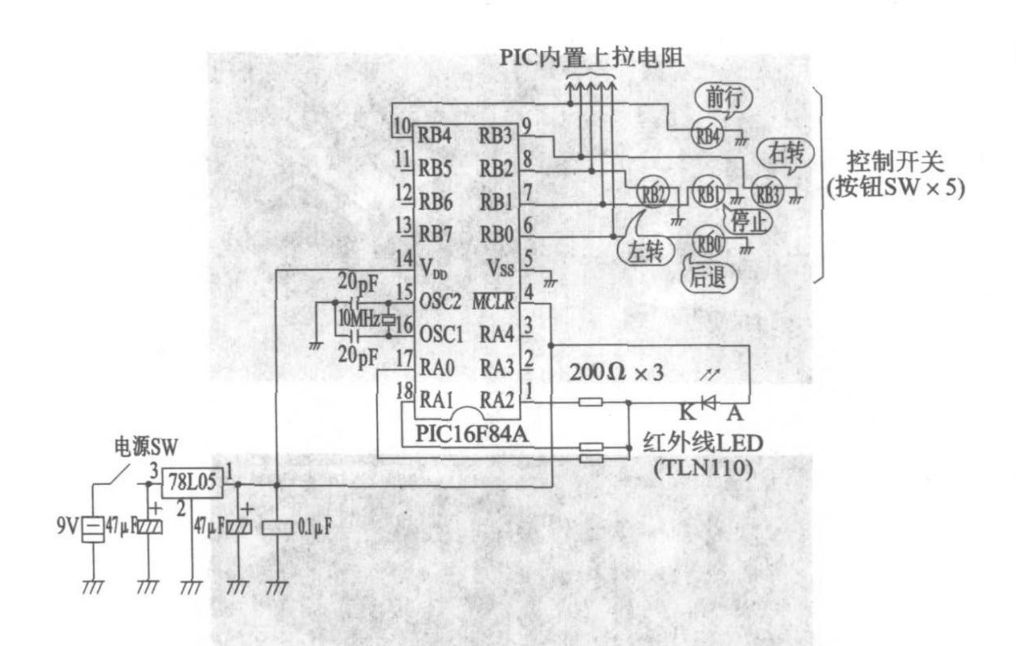


图 7.30　遥控坦克用发送器的电路

图 7.31 中标示出的 4 位动作数据应该与接收器一侧电机驱动芯片的输入信号相对应（参见表 7.5）。其中前面 2 位作为右电机，后面 2 位作为左电机的旋转控制信号。

图 7.31 发送数据格式

10位
开始位
动作数据
停止位
1 0 1 0 1 0
前进 10 10
后退 01 01
右转 00 10
左转 10 00
停止 00 00
右电机
左电机
10:正转
01:反转
00:停止

在遥控坦克中，由于电机的极性、安装取位、齿轮箱的对称性等原因，车轮（履带）的旋向会发生变化，应根据试验确认电机的转向和接线方式。

上面说过，遥控坦克采用 4 位动作数据，其实对于“前进、后退、右转、左转和停止”5 个动作只需要 3 位就可以表达了（3 位数据最大可以表示 $2^3=8$ 个动作）。

如果发送器传输的动作数据为 3 位，接收器程序也仍然可以将其转换（或者叫做译码）成 4 位动作数据，这样就能与电机驱动芯片的输入信号相对应了（参见图 7.32）。

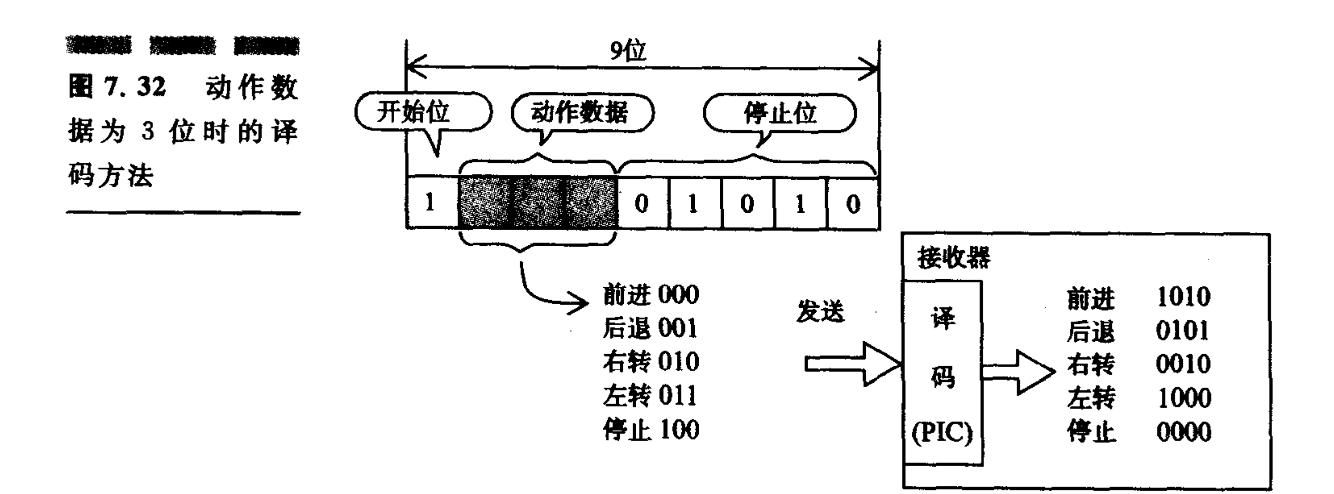


图 7.32 动作数据为 3 位时的译码方法

图 7.33 为遥控坦克发送器的照片，程序清单 7.3 为 PIC 遥控器编写的源程序。完工的发送器和干电池最好合装在一个合适的塑料盒内。

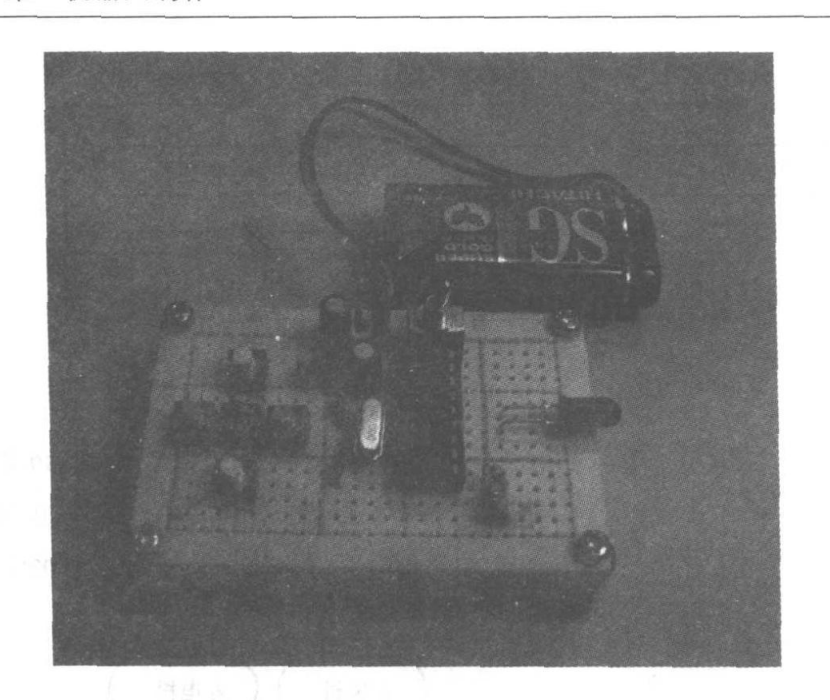


图7.33 遥控坦克发送器外观图

```
;程序清单 7.3
;红外线发送器源程序(TANK_TX)
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
        LIST        P=PIC16F84A         ;指定所使用的 PIC
        INCLUDE     "P16F84A.INC"       ;指定读取设定文件
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
LEDOFF  EQU         0FFH                ;设定 LED 熄灭数据
LEDON   EQU         00H                 ;设定 LED 点亮数据
CNT1    EQU         0CH                 ;定时器 1 的计数变量
CNT2    EQU         0DH                 ;定时器 2 的计数变量
CNT3    EQU         0EH                 ;定时器 3 的计数变量
WWW     EQU         0FH                 ;载波的计数变量
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
        ORG         0                   ;程序存储起始地址

        BSF         STATUS,RP0          ;选择存储单元第 1 列
        CLRF        TRISA               ;将端口 A 全部设定为输出模式
        MOVLW       0FFH                ;端口 B 设定用数据
        MOVWF       TRISB               ;将端口 B 全部设定为输入模式
        BCF         OPTION_REG,7        ;端口 B 的上拉电阻有效
        BCF         STATUS,RP0          ;选择存储单元第 0 列
```

```
        MOVLW   LEDOFF          ;将熄灯数据置入 W 寄存器
        MOVWF   PORTA           ;向端口 A 输出熄灯数据

;读取按钮数据 * * * * * * * * * * * * * * * * * * * * * * * * *
REPEAT  BTFSS   PORTB,4         ;检查 SW4
        GOTO    MAE             ;若为 0(ON)则跳转至前进
        BTFSS   PORTB,3         ;检查 SW3
        GOTO    MIGI            ;若为 0(ON)则跳转至右转
        BTFSS   PORTB,0         ;检查 SW0
        GOTO    USHIRO          ;若为 0(ON)则跳转至后退
        BTFSS   PORTB,2         ;检查 SW2
        GOTO    HIDARI          ;若为 0(ON)则跳转至左转
        BTFSS   PORTB,1         ;检查 SW1
        GOTO    TEISHI          ;若为 0(ON)则跳转至停止
        GOTO    REPEAT          ;重复

;发送各个按钮的数据 8 位(1 + ABCD + 10101) * * * * * * * * * * *
MAE     CALL    B_1             ;开始位 1
        CALL    B_1             ;数据 4 位 B1010
        CALL    B_0
        CALL    B_1
        CALL    B_0
        CALL    B_1             ;停止位 10101
        CALL    B_0
        CALL    B_1
        CALL    B_0
        CALL    B_1
        CALL    TIMER3          ;延时
        CALL    TIMER3
        GOTO    REPEAT

MIGI    CALL    B_1             ;开始位 1
        CALL    B_0             ;数据 4 位 B0010
        CALL    B_0
        CALL    B_1
        CALL    B_0
        CALL    B_1             ;停止位 10101
```

```
        CALL      B_0
        CALL      B_1
        CALL      B_0
        CALL      B_1
        CALL      TIMER3        ;延时
        CALL      TIMER3
        GOTO      REPEAT

USHIRO  CALL      B_1           ;开始位 1
        CALL      B_0           ;数据 4 位 B0101
        CALL      B_1
        CALL      B_0
        CALL      B_1
        CALL      B_1           ;停止位 10101
        CALL      B_0
        CALL      B_1
        CALL      B_0
        CALL      B_1
        CALL      TIMER3        ;延时
        CALL      TIMER3
        GOTO      REPEAT

HIDARI  CALL      B_1           ;开始位 1
        CALL      B_1           ;数据 4 位 B1000
        CALL      B_0
        CALL      B_0
        CALL      B_0
        CALL      B_1           ;停止位 10101
        CALL      B_0
        CALL      B_1
        CALL      B_0
        CALL      B_1
        CALL      TIMER3        ;延时
        CALL      TIMER3
        GOTO      REPEAT

TEISHI  CALL      B_1           ;开始位 1
        CALL      B_0           ;数据 4 位 B0000
        CALL      B_0
```

```
                CALL        B_0
                CALL        B_0
                CALL        B_1             ;停止位 10101
                CALL        B_0
                CALL        B_1
                CALL        B_0
                CALL        B_1
                CALL        TIMER3          ;延时
                CALL        TIMER3
                GOTO        REPEAT

;发送数据 0 * * * * * * * * * * * * * * * * * * * * * * * * * * *
B_0             MOVLW       D'50'           ;12×50 = 600μs
                MOVWF       WWW
                MOVLW       LEDOFF          ;将熄灯数据置入 W 寄存器
                MOVWF       PORTA           ;向端口 A 输出熄灯数据
L0              CALL        TIMER1
                DECFSZ      WWW,1
                GOTO        L0
                RETURN
;发送数据 1 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
B_1             MOVLW       D'25'           ;12×25×2 = 600μs
                MOVWF       WWW
L1              MOVLW       LEDON           ;将点亮数据置入 W 寄存器
                MOVWF       PORTA           ;向端口 A 输出点亮数据
                CALL        TIMER1
                MOVLW       LEDOFF          ;将熄灯数据置入 W 寄存器
                MOVWF       PORTA           ;向端口 A 输出熄灯数据
                CALL        TIMER1
                DECFSZ      WWW,1
                GOTO        L1
                RETURN

;时间子程序 * * * * * * * * * * * * * * * * * * * * * * * * * * *
TIMER1          MOVLW       D'6'            ;12μs 定时器子程序
                MOVWF       CNT1
LOOP1           NOP
                NOP
```

```
        DECFSZ    CNT1,1
        GOTO      LOOP1
        RETURN

TIMER2  MOVLW     D'62'       ;0.1ms 定时器子程序
        MOVWF     CNT2
LOOP2   NOP
        DECFSZ    CNT2,1
        GOTO      LOOP2
        RETURN

TIMER3  MOVLW     D'100'      ;10ms 定时器子程序
        MOVWF     CNT3
LOOP3   NOP
        CALL      TIMER2
        DECFSZ    CNT3,1
        GOTO      LOOP3
        RETURN

        END                   ;程序结束
```

程序清单 7.3　遥控坦克的发送程序

7.2.3　遥控坦克的接收器

图 7.34 给出遥控坦克接收器电路，该电路与第 6 章图 6.12 相比，仅将接在输出端口 RB0～RB3 的 4 个 LED 改为 2 个电机驱动芯片（东芝 TA7257P）。如表7.5所示，TA7257P 芯片能够从 2 位输入端口上读取的数据控制 DC 电机正转、反转、停止或制动。

表 7.5　用电机驱动芯片（TA7257P）控制电机旋转

输　入		输　出		动　作
IN1	IN2	OUT1	OUT2	
0	0	高阻抗		停止
0	1	L	H	正转/反转
1	0	H	L	反转/正转
1	1	L	L	制动

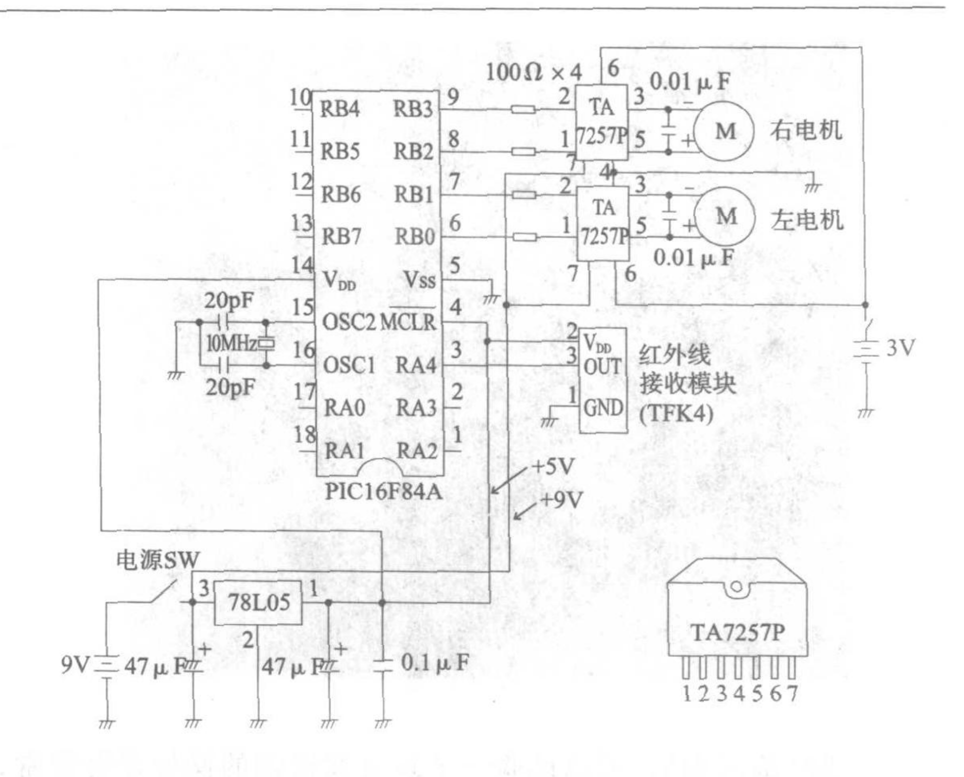


图 7.34 遥控坦克的接收电路

遥控坦克接收器从 3 引脚稳压器芯片(78L05)取得 5V 供电而运行,但是 TA7257P 的工作电压为 6～18V,所以这个芯片的电源(引脚 7)应直接从 006P 干电池取得。推土机 DC 电机的额定电压为 3V,所以需要 2 节 5 号干电池串联。系统中两种电池的揿钮相同,在接线时切勿弄错。电源开关使用了双联拨动开关,两个电源可以同时开闭。

为 PIC 编写的接收程序与第 6 章程序清单 6.2 完全相同。图 7.35 为接收器电路板的外观照片。

接收器面包板完工后,应将其连同电池一道安装在坦克机座上。在图 7.36 中,作者将坦克的机座部分加工平整安装约 7cm×9cm 的面包板,再用螺钉将接收器的电路板和电池盒固定其上。

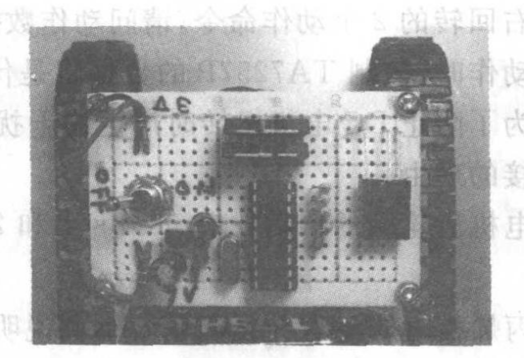


图 7.35 接收器电路板的照片

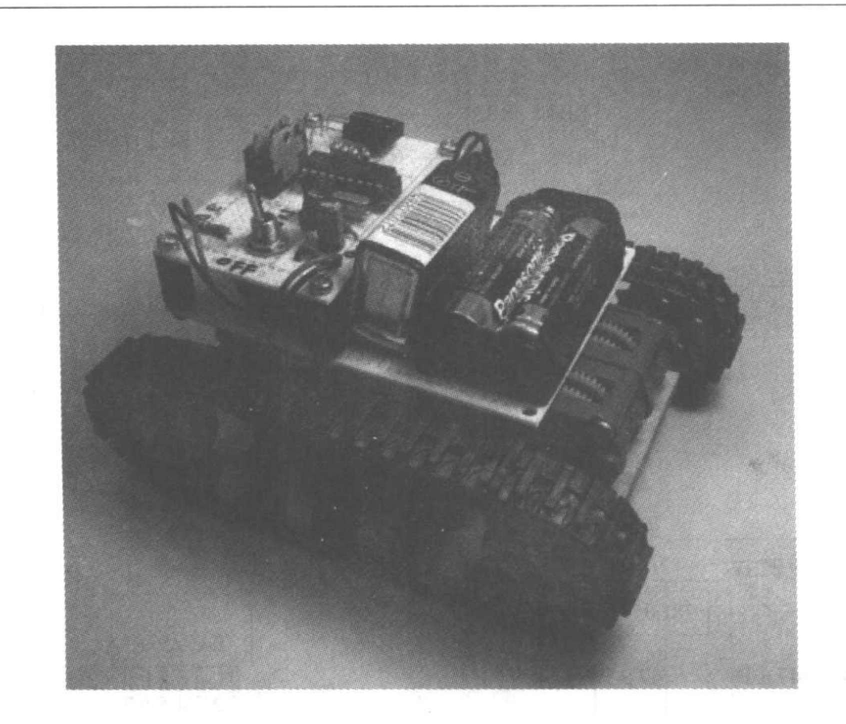


图 7.36　搭载接收器电路板和电池的坦克机座

上述准备完成后，可以试验一下坦克发送器的操作是否正常，注意应该将发送器的红外线 LED 对准接收器的接收模块。

如果想让遥控坦克运行更为平稳、马力更为强劲，必须在减轻重量和选用强力电机上下一番功夫。请先以本节这个遥控坦克为基础，再去挑战你心目中憧憬的遥控坦克吧。



练习题

1. PIC16F84A(20MHz)和 TA7257P 的工作电压范围分别为多少？采用两种芯片组成的电路是否可以使用同一种电源？
2. 红外线遥控坦克具有前进、后退、右转、左转、停止等功能，如果需要增加原地左右回转的 2 个动作命令，请问动作数据最少需要多少位？做原地左右回转动作时，控制 TA7257P 的数据将是什么？
3. 说明为了防止 DC 电机运转时产生的干扰噪声，电容应尽可能靠近电机端子连接的理由。
4. 双联电机齿轮箱的传动比有 58.2：1 和 203.7：1 两种，哪一种的转矩较大？
5. 加工丙烯板时应注意什么问题？举例说明。
6. 简要说明光电判读器检测黑色线条的原理。

第8章

机器人比赛集锦

日本人长久以来对机器人怀有特殊的亲切感情，现在的年轻一代是在以铁臂阿童木为代表的机器人动画片的熏陶下长大起来的。今天双足步行机器人频频闪亮登场，使得日本的机器人热有增无减。

究其原因，在于日本国民与生俱来的对技术抱有的浓厚兴趣，即所谓的“技术立国”。

在这种背景下，日本广泛开展了普及性和国际性的机器人大赛，例如相扑机器人大赛、RoboCup、救援机器人大赛等等，对象之广泛、主题之丰富在国际上也实属罕见。

本章就各种大赛的概要和特点进行简单的说明，有兴趣的读者务必亲自尝试一下这些机器人制作的挑战。

8.1　机器人相扑大赛

第 1 届机器人相扑大赛于 1990 年在日本召开，到 2003 年已经是第 15 届了(本届的主办者是**富士软件 ABC 株式会社**，以及与高中教育有关的部门——**全国职业高中校长协会**)。

机器人相扑大赛的主页上公开发布了举办这个活动的宗旨，是“通过机器人制作为学生以及社会青年提供一个学习基本技术技能、提高研究兴趣和发挥创造力的场所，同时享受“创造”的乐趣。

每年日本从 8 月到 10 月期间在全国范围内举办分区比赛，10 月进行高中组全国大赛(从第 5 届开始实施)，年底进行包括各个年级级别的一般性全国大赛(地点在东京)。比赛初期赛事的规模较小，全国只有数百台机器人参赛，自从高中学生参赛之后，声势逐渐扩大到几千台的规模。2003 年起在 3kg 级的基础上增加了 10kg 级。从 1998 年的第 10 届大赛起开始出国远征，从第 13 届起开始试行以美式足球为参考原型的机器人足球赛。

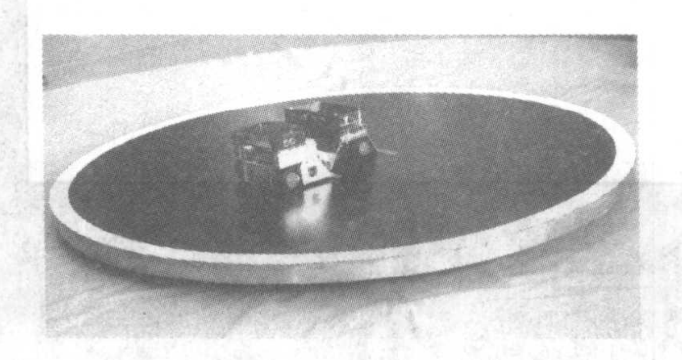


图 8.1　相扑机器人大赛的场景(3kg 级)

相扑机器人大赛，顾名思义，比赛方式参考了相扑的有关规则，让机器人力士扮演实际相扑选手在摔跤场中进行角力对抗，以将对方推出摔跤场外的方式决出胜负(参见图 8.1)。相扑机器人大赛分搭载微控制器的自主型和通过遥控操作的无线电控制型两种。表 8.1 为机器人的规格，图 8.2 为相扑机器人的实物照片。

表 8.1　相扑机器人的规格

	3kg	10kg
外　形	・宽 20cm ・长 20cm ・高度任意	・宽 30cm ・长 30cm ・高度任意
重　量	・包含所有附属零部件总共 3kg 以下。无线电控制型不包括比例遥控系统的重量 ・10kg 级的重量为 10kg 以下	
自主型的动作开始方法	・动作开始从按下动作开关 5 秒钟之后起算，所以事先应做好充分准备 ※无线电遥控型听从主裁判的开始指令后开始比赛	
自主型的停止方法	・为了防止危险发生，机器人安装了遥控操作的“停止装置” ※无线电遥控型由遥控器操作	

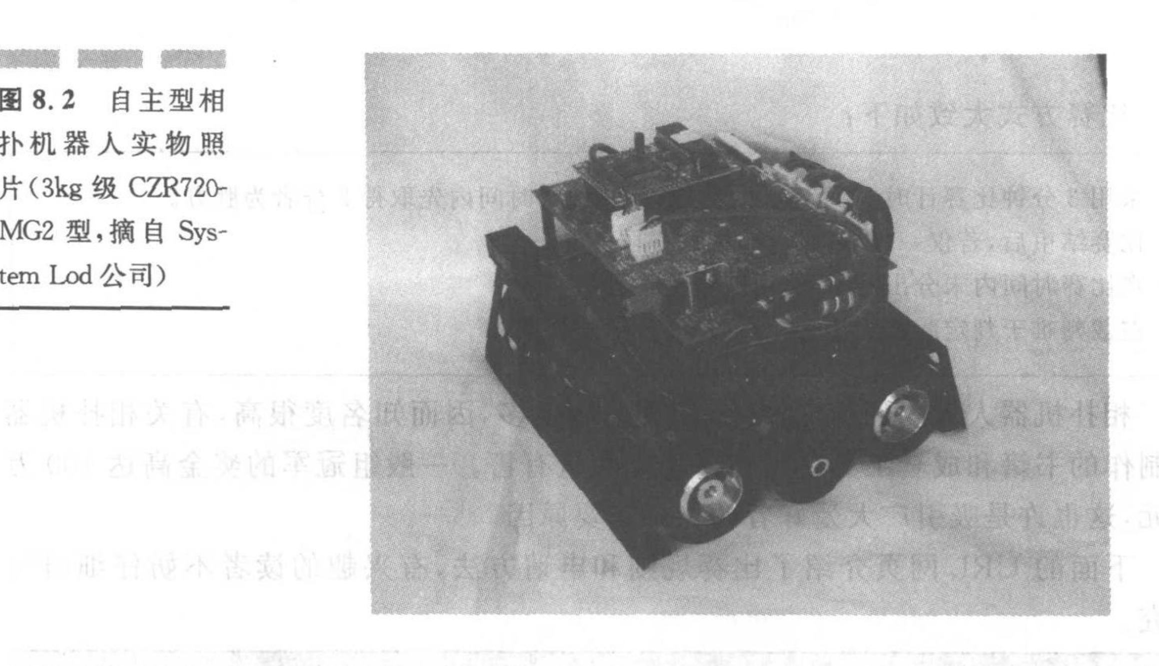


图 8.2　自主型相扑机器人实物照片(3kg 级 CZR720-MG2 型，摘自 System Lod 公司)

相扑机器人比赛摔跤场地的规格见表 8.2，以 3kg 级的摔跤场地为例，如图 8.3所示。

表 8.2　相扑机器人的比赛摔跤场地规格

	3kg 级	10kg 级(分 4 块组成)
直　径	154cm	227cm
材　质	冷轧钢板	
颜　色	・场地内……黑色 ・出发线……茶色(宽 2cm,长 20cm) ・边界线……白(宽 5cm)	・场地内……同左 ・出发线……茶色(宽 2cm,长 30cm) ・边界线……白(宽 10cm)
场地亮度	不超过 1000lx	

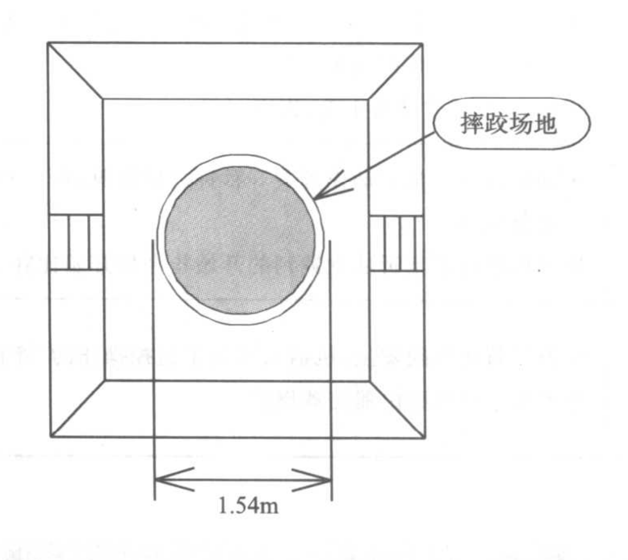


图 8.3　相扑机器人的摔跤场地(3kg 级)

比赛方式大致如下：

・采用 3 分钟比赛计时,3 局定胜败的原则,在比赛时间内先取得 2 分者为胜方。
・比赛结束后,若仅一方取得 1 分,则该方取胜。
・在比赛时间内未分出胜负时,由裁判判定胜负。
　在裁判难于判定胜负的场合,可加赛 3 分钟延长期。

相扑机器人大赛的历史较长,参赛的队也多,因而知名度很高,有关相扑机器人制作的书籍和成套工具材料在市场上大量有售。一般组冠军的奖金高达 100 万日元,这也许是吸引广大爱好者的一个重要原因。

下面的 URL 网页介绍了比赛规则和申请方法,有兴趣的读者不妨仔细研究研究。

全日本相扑机器人大赛公开网址…………http://www.fsi.co.jp/sumo/

8.2 遥控赛车比赛

遥控赛车比赛于 1996 年举办第 1 届，2003 年迎来了第 8 届赛车大会（主办者为全国职业高中校长协会和北海道职业高中校长分会）。

在该大赛网页上公布的大赛宗旨是“通过遥控赛车的比赛，学习机械电子的基本技术技能，培养自发的、创造性的学习态度，借助学生动手参与制作的手段推行课题研修式的教育改革，达到培育创新意识的目的。”

2003 年大赛参加者约有 2000 台，是巡线小车大赛中规模最大的一次。比赛的赛程是在每年的 10 月到当年年底进行分区赛，翌年元月在北海道举行全国大赛。

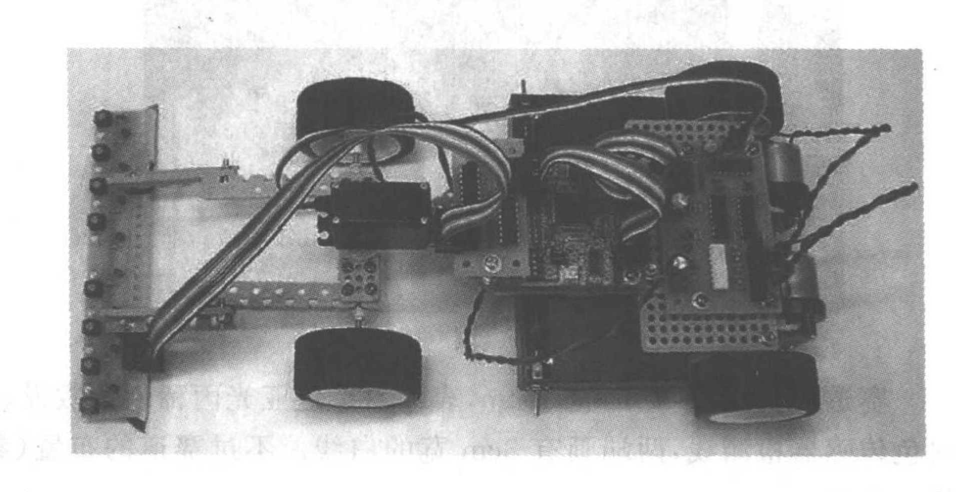


图 8.4 遥控赛车的实例照片（摘自株式会社文化总研：http://www.bun-net.co.jp）

遥控赛车比赛的方式是由大会提供微控制器，由参赛者将它搭载到自制的赛车上，并下载自己编制的原始程序，然后运行小车，看谁能更快地走完规定的路线。

大赛分高中组和一般组，高中组规定使用由大会提供的 CPU 以及其它零部件，报名方法也与一般组不同，以学校为单位申请报名。一般组的参加者第一次报名时，必须缴纳包含购买 CPU 主板在内的参赛费。大会提供的 CPU 主板统一采用精工社生产的 H8 系列产品。

有关制作的步骤和历届冠军赛车的信息均在公开网页上有详细记载。

竞赛规则的条款中与赛车本体制作有关的要求大致如下。

- 电源和能源必须是 5 号碱性电池和 5 号充电电池（1.2V），且在 8 节以下。
- 赛车的外形尺寸必须小于宽 300、高 150，但重量和材质不限。
- 不许采用具有吸引功能的赛车。

有关竞赛的基本规则如下：

- 一场比赛由 2 台赛车同时行走。
- 从同一赛道的内侧（IN）和外侧（OUT），在“红・红・红・绿”的信号灯点亮

后同时发车，并经由立交赛道穿行。

- 在同一赛道上，两台车错开半圈的距离，看起来好像两台车在进行追逐赛。
- 如果后面的车追上前面的车，被追上的一方必须遵从裁判的指令给后面的车让道。
- 比赛用完成赛段花费的时间决出胜负。

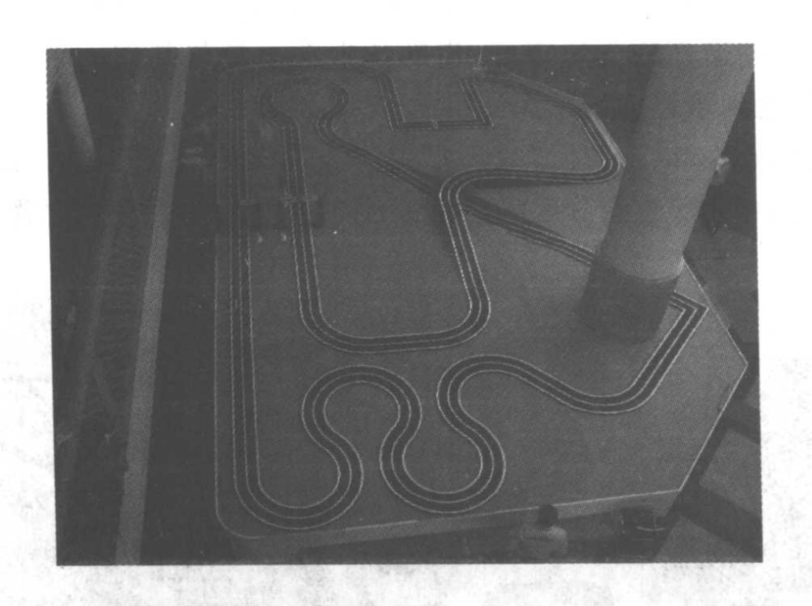


图 8.5　遥控赛车的赛道照片

赛道的长度约 50m，宽 30cm，表面由黑色亚光丙稀软片敷设，中间有 2cm 宽的白色传感器检测线，两端画有 3cm 宽的白线。不过赛道的布置(参见图 8.5)是年年改变的。

下面的 URL 网页介绍了比赛规则和申请方法，读者不妨仔细研究研究。

遥控车赛公开网址………………http://www.mcr.gr.jp/

8.3　微型机器鼠大赛

第 1 届微型机器鼠大赛要追溯到 1980 年。到微型机器鼠 2003(2003 年)已经是第 24 届了，是机器人比赛中历史最长赛事(主办由财团法人：新技术振兴财团)。

在这项赛事中，让机器人(本节称之为微型机器鼠)穿行迷宫，角逐它们的智能程度和行走的速度。赛事的赛程安排在每年 11 月，进行高级组的预选赛和正式比赛，以及面向初学对象的新人组比赛。不过这个赛事近年来人气每况愈下，参赛数量竟未超过百台，但是它毕竟是历史最长，水平较高的机器人竞技比赛之一。

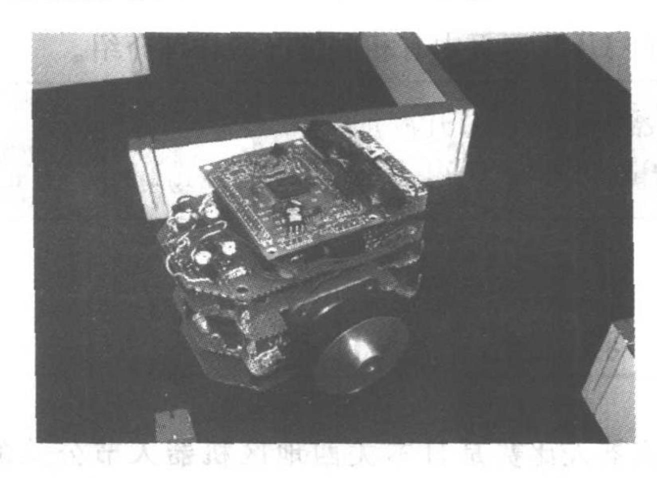


图 8.6 微型机器鼠实物照片

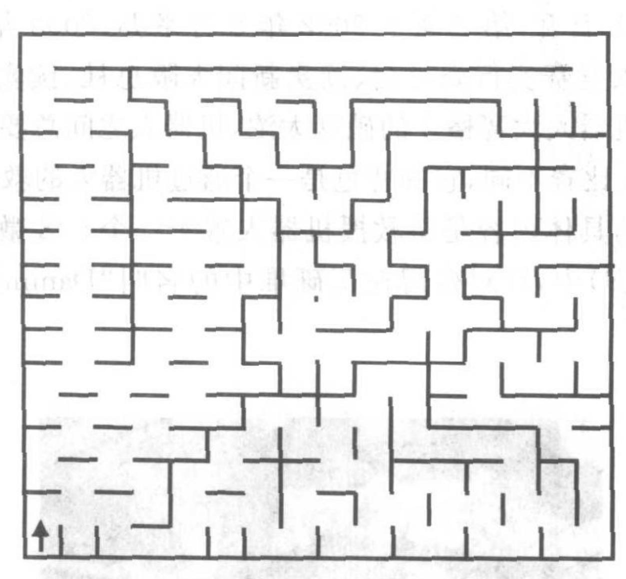


图 8.7 微型机器鼠迷宫的图案

微型机器鼠大赛规则的主要内容和有关迷宫的规定如下。图 8.7 是一个迷宫的实例。

本体

- 自主型,不允许使用燃烧型的能源。
- 禁止跨越、攀爬、损坏、破坏迷宫的隔墙。
- 外形尺寸以向地面的投影为限,不应超过边长为 25cm 的正方形。若在行进中改变形状,也不允许超过这一限制,但是高度不限。

迷宫

- 迷宫隔墙的侧面为白色,顶面为红色,地面为黑色。
- 迷宫由若干 18cm×18cm 的单元组成,整个场地有 16×16 个单元。隔墙的高为 5cm,厚度为 1.2cm。
- 迷宫的起点可以设在正方形四个角中的任意一处,按顺时针方向出发,终点为正中间的四个小单元。

下面的URL网页中有相关情况的详细介绍。

微型机器鼠大赛的公开网址……

http://www.bekkoame.ne.jp/～ntf/mouse/mouse.html

8.4 救援机器人比赛

救援机器人比赛是日本关西地区**机器人节**公认的竞技项目，第1届比赛于2001年7月召开，第2届于2002年8月举办，2003年是第3届大赛（主办单位为救援机器人竞赛实行委员会、读卖新闻大阪总社、读卖电视台等）。

以往机器人大赛格斗的硝烟太浓，机器人之间总要在竞争或者对战中一决高低。救援机器人比赛不同，它的特色是一个通过机器人的救援活动给世人启示的比赛。

比赛的具体内容是将救援机器人置于一个1/8微缩比例的地震灾区模拟场地（参见图8.9）中，针对残留在瓦砾堆中的名叫“Dammy Yan”的模拟遇难者进行救助比赛。

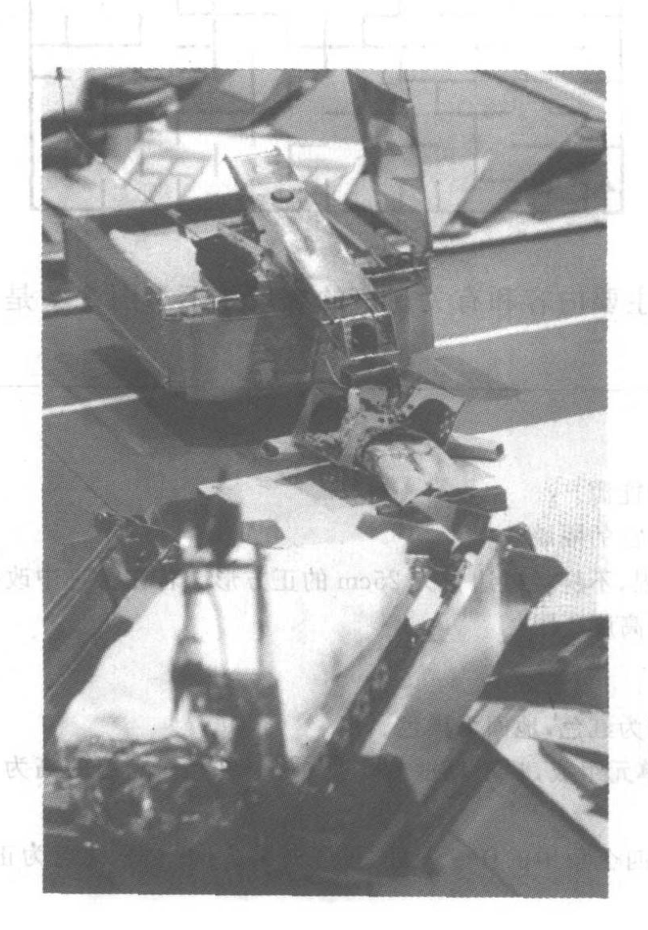


图8.8 救援机器人比赛实例场景（摘自救援机器人竞赛实行委员会）

救援机器人比赛的参赛资格没有特别的限制，参赛者可以是高中、职高、大学以及一般群众，参赛人员基本上是自编成队，事先学习参考书籍。参赛人员还可以接受赞助。

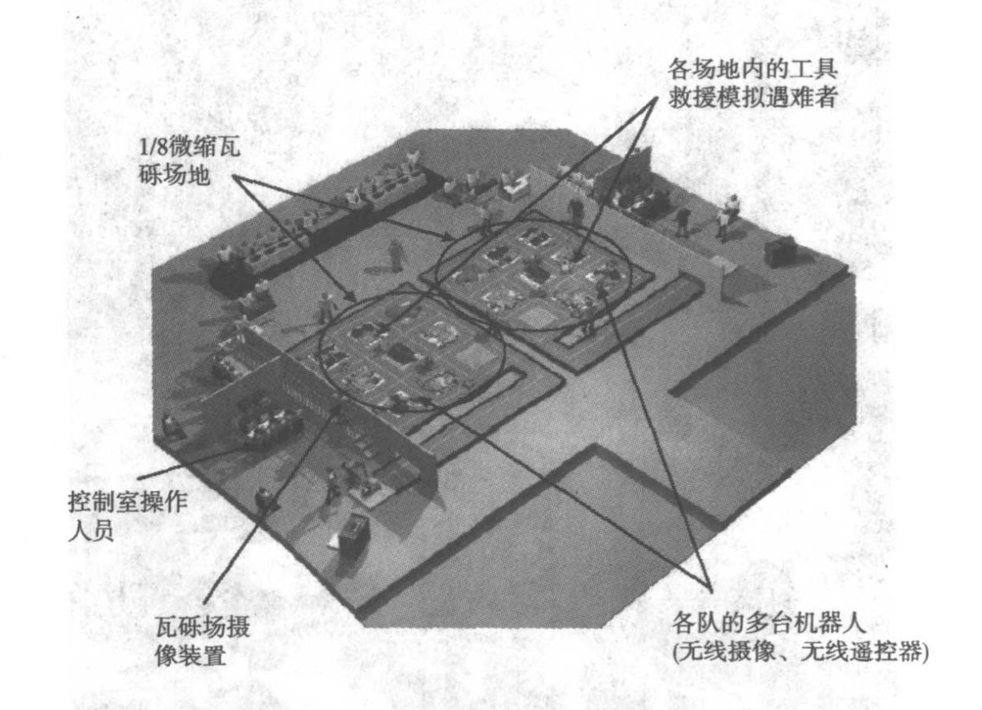


图 8.9 救援机器人大赛场景（摘自救援机器人竞赛实行委员会）

救援机器人比赛规则未对机器人的大小做出明确规定，不过在救援活动前，机器人必须能容纳于 1200mm×1200mm 的出发区内，出发时必须经由一条宽 1200mm×高 450mm 的通道。机器人车体搭载无线摄像头，操作人员不允许直接观察场地，而必须借助摄像头返回的画面进行无线遥控操作。

救助对象“Dammy Yan”（2003 年大赛中体重为 850g）不仅仅是一个偶人，而且装有体表传感器、拉力传感器和加速度传感器等等，若施加的力或冲击过大将被扣分。竞技内容包括清理瓦砾和障碍物、搜索救援对象、细心地救护遇难者并转移到安全地点等等。

2003 年大赛的场地设置情况如下：

· 瓦砾中设置了数具“Dammy Yan”的遇难者。

· Dammy yan 的设置地点可以通过目测确认，有时也可能被埋藏在其它地点。由于瓦砾堆放在私有地产上，不能随意破坏或强行通过。

· 由于担心二次灾害，所以人不能干预。

可见，这是一项与一般机器人比赛风格迥然不同，内容非常独特的机器人竞赛。

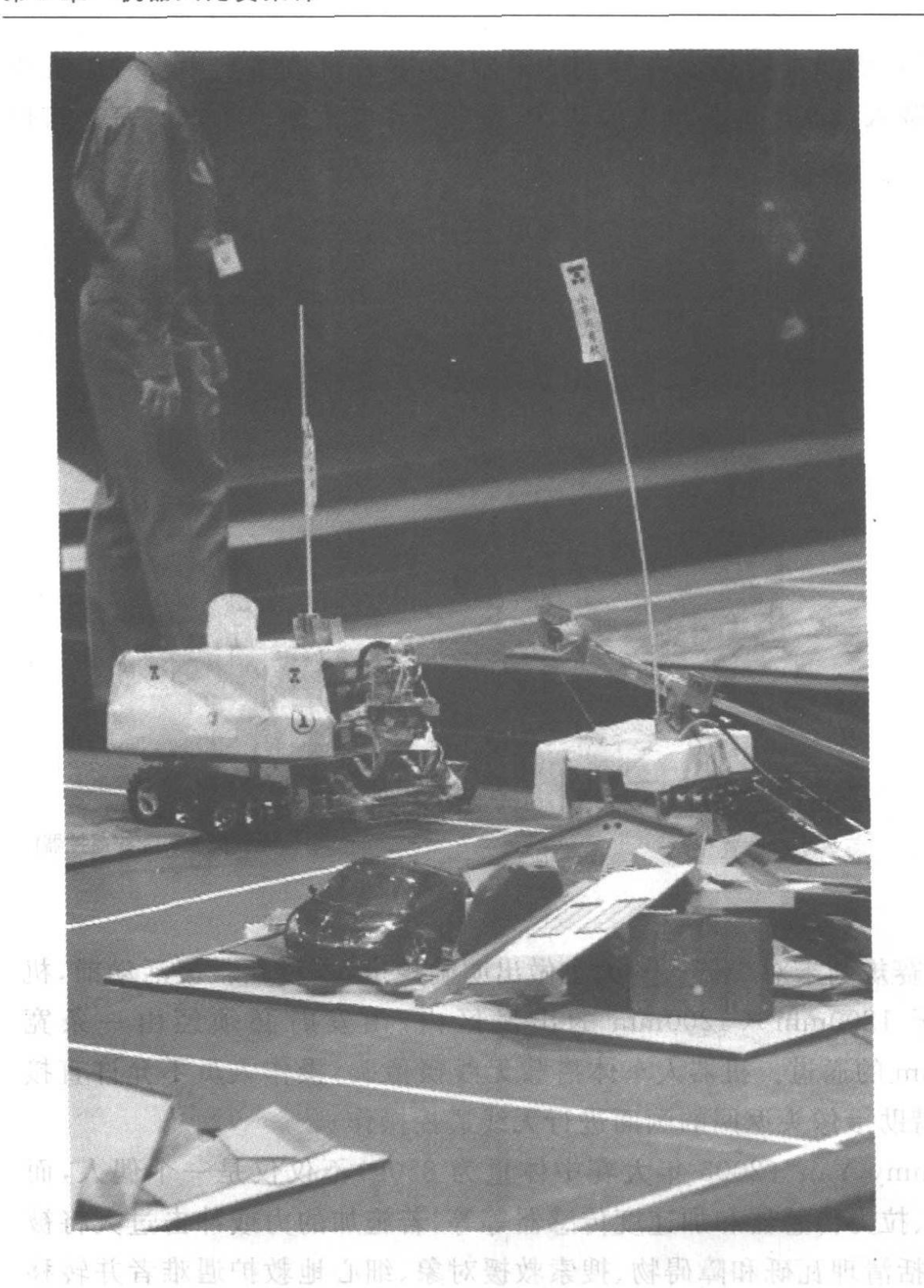


图 8.10　救援机器人大赛的比赛场景（摘自救援机器人竞赛实行委员会）

从下面的 URL 网页可以了解详情。

救援机器人大赛的公开网址……http://www.rescue-robot-contest.org/

8.5 RoboCup 大赛

RoboCup 从 1992 年开始筹备，1997 年第 1 届 RoboCup 世界大赛召开，分仿真组、小型组、中型组 3 个级别进行比赛。自 1998 年起举办日本全国大赛，起名为“开放的日本”。图 8.11 表示小型机器人组竞赛的场景。

RoboCup 设定了一个宏伟的目标，就是到 2050 年组建一支拟人机器人球队战胜 FIFA 的冠军。真酷!!（主办者为东京都 NPO RoboCup 日本委员会。）



图 8.11 小型机器人组比赛场景（摘自 NPO RoboCup 日本委员会）

当年，经日本众多学者的大力提倡，RoboCup 创办成了一项以足球为题材的自主移动机器人赛事。现在，这一项赛事的概念已经拓展开来，成为不仅针对足球，而且涉猎大规模灾害救援、培育未来技术人才等领域的组织。

作为 RoboCup 最初的机器人足球世界杯大赛，在 2000 年就形成自主移动机器人 3 个组和仿真机器人 1 个组，共 4 个组别的赛事。这些组别的概况简单介绍如下：

1. 小型机器人组

RoboCup 小型机器人组的场地尺寸为 152.5cm×274cm（大致相当于乒乓球台），小型自主移动机器人的直径在 18cm 以下，各队由少于 5 台的机器人进行编队分组，使用橙色的高尔夫球进行对抗比赛。

规则规定：参赛的机器人在地面的投影面积不超过 180cm^2，且能够放入直径为 18cm 的圆筒内，高度不超过 15cm（带摄像头时不超过 22.5cm）。

2. 中型机器人组

RoboCup 中型机器人组的场地大小相当于 9 个乒乓球台，机器人的直径不超过 50cm，高度在 30cm～80cm 之间，每个队由 4 台机器人组成，采用橙色足球进行对抗比赛。



图 8.12　中型机器人组比赛场景（摘自 NPO RoboCup 日本委员会）

3. *仿真机器人组*

在计算机模拟的假想场地上，通过程序设计让虚拟运动员进行足球比赛。是 RoboCup 活动开展最早的一个组别。

4. *4 足步行机器人组*

采用 SONY 制作的玩具机器人进行 3 对 3 的足球比赛。遗憾的是机器人只借给选拔队参加比赛，一般群众不能参加。

RoboCup 的详细介绍见下面 URL 网页。

RoboCup 日本委员会的公开网址…………http://www.robocup.or.jp/

8.6 ROBO-ONE 比赛

ROBO-ONE 是针对双足步行机器人的格斗大赛（由 ROBO-ONE 执行委员会“BestTechnilogy 株式会社”主办，日本科学未来馆协办）。在商业广告中广为百姓熟知的双足步行机器人有本田公司的 ASIMO、索尼公司的 QRIO 等，它们都表现出很强的亲和力。

由于这样的广告效应，尽管这项赛事的历史不长，但是在第 3 届比赛上报名竟多达 93 台，其中且不乏高水平的机器人，现在 ROBO-ONE 已经成为令人瞩目的机器人大赛。第 1 届 ROBO-ONE 比赛于 2002 年 2 在日本科学未来馆进行，同年 8 月在川崎产业振兴会馆又举行了第 2 届比赛。图 8.13 所示为冠军争斗战的情景。

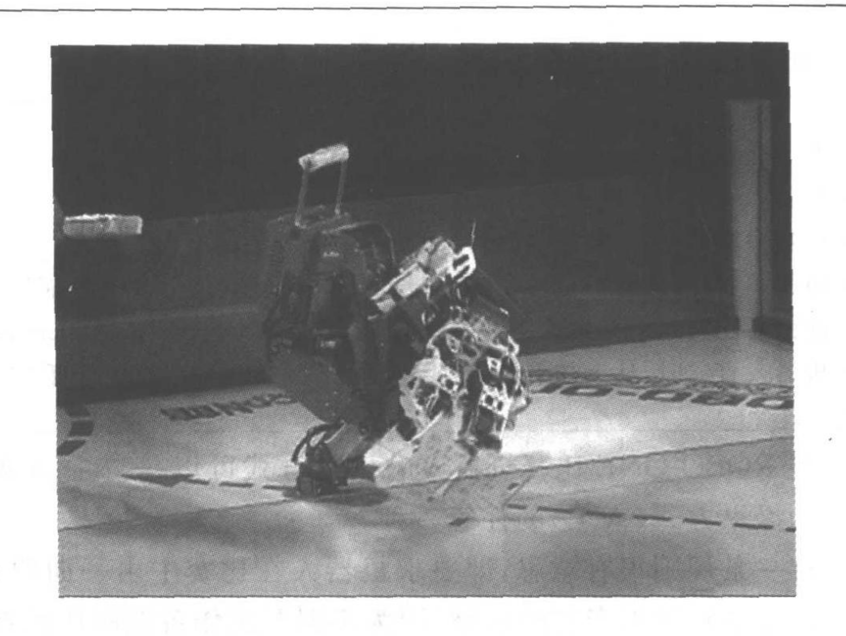


图 8.13 第 3 届 ROBO-ONE 冠军战的场景（摘自 NPO RoboCup 日本委员会）

ROBO-ONE 竞赛的特点是内容每年都有所不同，本节简单地介绍一下在本书即将脱稿时最新的第 4 届比赛的有关内容。

有关参赛资格的内容如下：

- 符合 ROBO-ONE 竞赛规则。
- 双足步行机器人在 20 秒钟内能够走 5 步以上，行走时脚掌离地。
- 躯体能够伸屈。
- 能够走横步，行走时脚掌离地。

比赛方式为：参赛者独立制作的机器人在事先给定的场地（参见图 8.14）范围内做公开表演和比赛，最后由裁判来判定胜败。

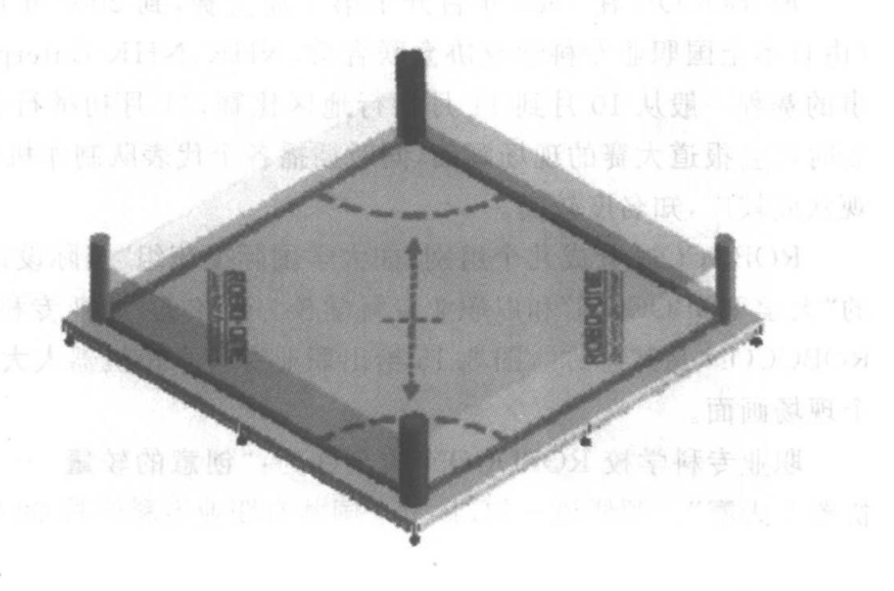


图 8.14 第 2 届 ROBO-ONE 的比赛场地（摘自 NPO RoboCup 日本委员会）

对于机器人规格的规定如下：

- 必须是双足步行机器人。
- 机器人身高为 20～120cm。
- 机器人脚的形状和样式在不抵触如下要求的条件下可以任意。
- 将脚掌(接触地面的部分)的最大长度设定为腿以下的长度。而所谓腿，指从躯干分离处的前后左右运动轴到地面的高度。不过脚掌最大长度限于 20cm 内。
- 机器人直立时，从上往下看，左右脚掌的最大轮廓连线相互不重叠。

在 ROBO-ONE 比赛机器人的操纵方式可以是自主操纵方式(计算机控制)，或手动操纵方式(人工)的任意一种。

有一条规则很有意思，即要求机器人在比赛中由一曲原创的主题歌伴奏。这是一件关系到知识产权的事情，因为不但要求作者能制作机器人，还要有作曲的才能。这一条也许成了实施比赛意想不到的瓶颈。

ROBO-ONE 比赛的内容更为广泛，不仅有同台较量，而且还有上下楼梯、开关门等竞赛科目。不仅有双足步行机器人比赛，同时还设有计算机仿真模拟比赛组。下面的 URL 网页登载了 ROBO-ONE 详细的介绍，有兴趣的读者可以登录相应的网站。

ROBO-ONE 的公开网址………………http://www.robo-one.com/

8.7 职业专科学校机器人大赛 ROBOCON

ROBOCON 在 1988 年召开了第 1 届大赛，到 2003 年已经连续举办了 16 届(由日本全国职业专科学校协会联合会、NHK、NHK Enterprise21 共同主办)。赛事的赛程一般从 10 月到 11 月举行地区比赛，11 月初举行全国大赛。由于 NHK 届时都会报道大赛的现场画面，同时插播各个代表队制作机器人的过程，所以它的观众面较广，知名度较高。

ROBOCON 分成几个组别，如大学国际交流组“国际设计比赛 IDC”、面向国内的“大学 ROBOCON”和以职业专科学校为对象的“职业专科学校 ROBOCON”等。ROBOCON 是其总称。图 8.15 给出职业专科学校机器人大赛 ROBOCON 组的一个现场画面。

职业专科学校 ROBOCON 的全称为：**“创意的较量——全日本职业专科学校机器人大赛”**。顺便说一句，日本全国共有职业专科学校 62 所。

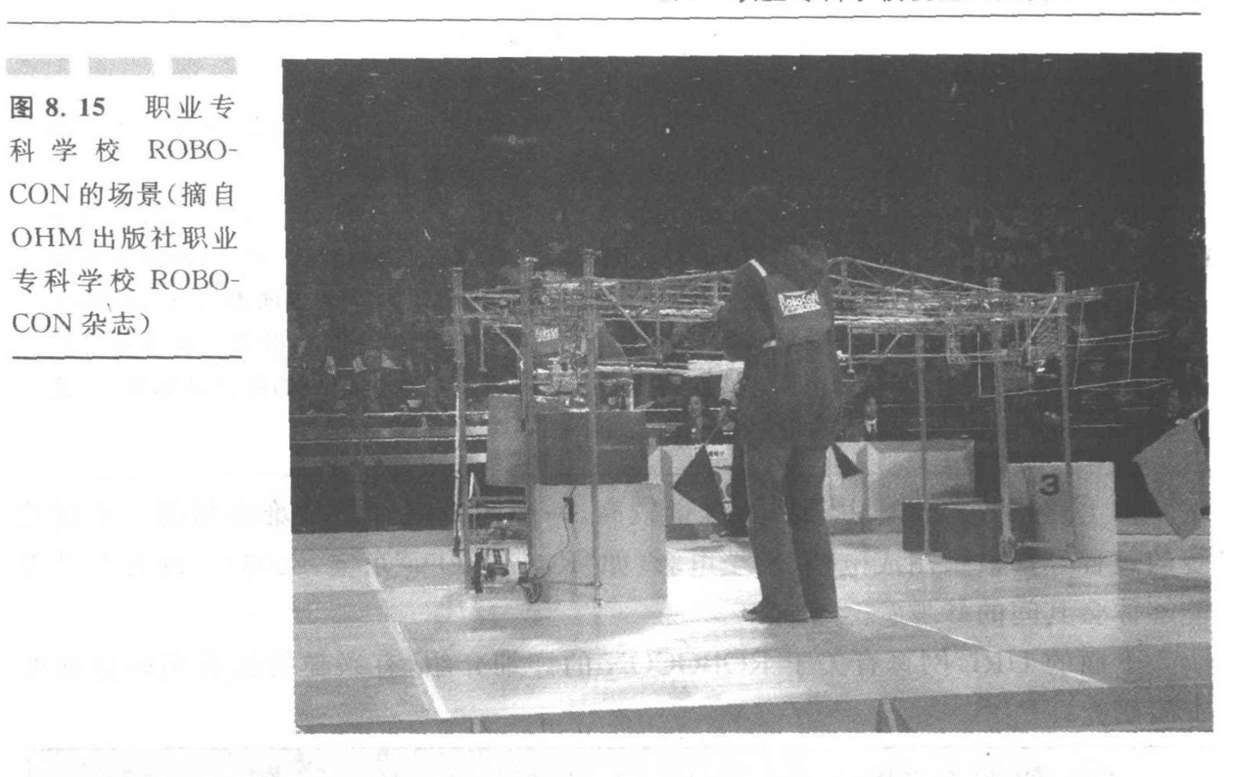


图 8.15 职业专科学校 ROBOCON 的场景（摘自 OHM 出版社职业专科学校 ROBOCON 杂志）

ROBOCON 每年都重新设定新的主题，比赛的内容就是发挥队员的创造性来制作构思的机器人，并通过竞赛检验自己的创新成果。

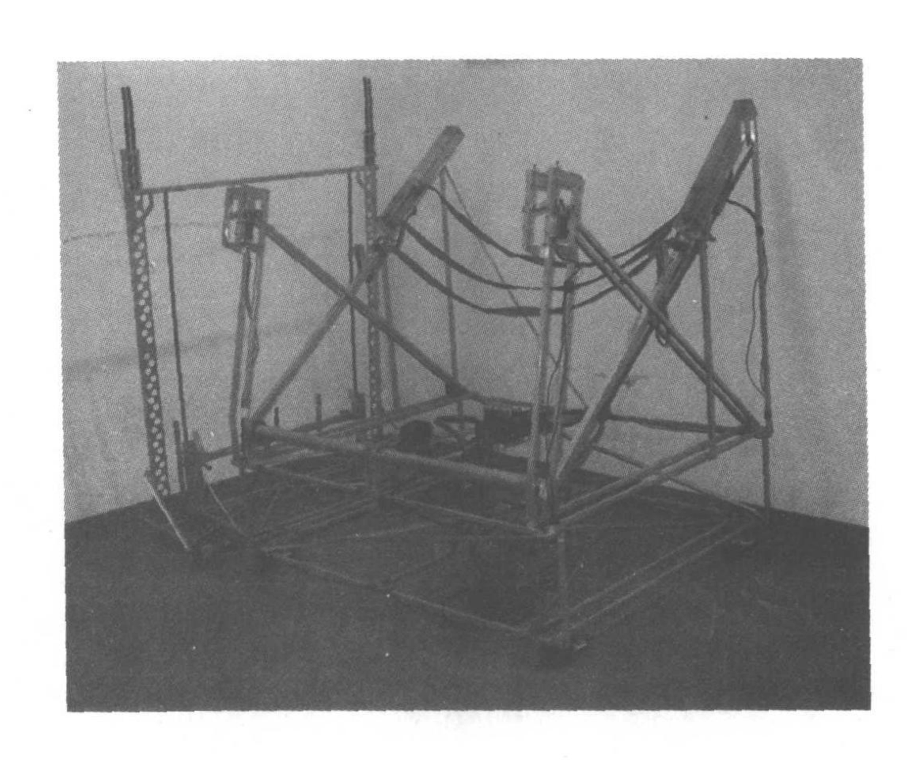


图 8.16 创意的机器人实例（明石职业专科学校：摘自机器人工程研究会）

例如，第 1 届大赛的主题是“干电池小车速度大赛”。这个题目的与众不同之处是用两节干电池做能源，驮上体重 60kg 以上的人，在 35m 长的赛道上行驶，看

谁的速度更快。

随着 ROBOCON 比赛的进程，实际上，历年比赛的内容都有所区别，难度在逐渐提高，例如已经在网上公布的 2003 年大赛的主题如下：

> 比赛场地上设置了转动的磨盘，还备有球体和立方体的目标物体。红、蓝两队遥控操作机器人尽力地将目标物体堆积起来，以便目标物体自身的重力让磨盘向着标有本队颜色的一侧旋转。总共设置了 3 处旋转磨盘，从正中分别用红、蓝两色各半分割。得分的计算方法是将每一个旋转磨盘旋至本方一侧得 1 分。比赛完成后根据得分的高低计算胜负，比赛的时间为 3 分钟。

像上述形式的比赛由于技术含量较高，一般的爱好者是很难参与的。不过比赛的过程经常被录制成电视、甚至电影(如“ROBOCON”东宝，2003)。读者们是否能够体会其间的妙趣?

下面的 URL 网页有关于 ROBOCON 的详细介绍，有兴趣的读者不妨仔细地查阅研究。

ROBOCON 公开网址………………http://www.official-robocon.com/

附　　录

第 3 章　程序清单 3.1

```
;* * * * * * * * * * * * * * * * * * * * * * * * * * * *
;程序清单 3.1
;直流电机控制程序
;* * * * * * * * * * * * * * * * * * * * * * * * * * * *
            LIST        P = PIC16F84A    ;指定使用的 PIC
            INCLUDE     "P16F84A.INC"   ;指定读入设定文件
;* * * * * * * * * * * * * * * * * * * * * * * * * * * *
WORK        EQU         0CH             ;保存转动数据
CNT1        EQU         0DH             ;定时器 1 的计数变量
CNT2        EQU         0EH             ;定时器 2 的计数变量
CNT3        EQU         0FH             ;定时器 3 的计数变量
MDIR        EQU         10H             ;转动方向的变量
 * * * * * * * * * * * * * * * * * * * * * * * * * * *
            ORG         0               ;存储程序的初始地址

            BSF         STATUS,RP0      ;选择存储单元第 1 列
            MOVLW       0F0H            ;把 11110000B 代入寄存器 W
            MOVWF       TRISB           ;设定端口 B 的上位为输入模式,下位
                                        ;为输出模式
            BCF         STATUS,RP0      ;选择存储单元第 0 列
            CLRF        PORTB           ;端口 B 清零(电机停止)
            MOVLW       01H             ;W←00000001B(正转)
            MOVWF       MDIR            ;设原转动方向为正转。

LOOP        MOVLW       01H             ;W←00000001B (正转)
            BTFSC       PORTB,5         ;检查 RB5
            MOVLW       02H             ;若为 1(ON),W←00000010B(反转)
            MOVWF       WORK            ;保存转动数据
            CALL        CKMDIR          ;检查转动方向
            MOVWF       PORTB           ;将转动数据输出至端口 B
```

```
        BTFSS    PORTB,4       ;检查 RB4
        GOTO     LOOP0         ;若为 0(OFF),继续转动

        MOVF     WORK,0        若为 1(ON),间歇转动读出转动数据
        MOVWF    PORTB         ;将转动数据输出至端口 B
        CALL     TIMER3        ;调用 1s 定时器
        MOVLW    03H           ;W←00000011B (制动)
        MOVWF    PORTB         ;将转动数据输出至端口 B
        CALL     TIMER3        ;调动 1s 定时器

LOOP0   MOVF     WORK,0        ;读出转动数据
        MOVWF    MDIR          ;保存转动数据
        GOTO     LOOP          ;循环
                               ;转动方向检查子程序
CKMDIR  SUBWF    MDIF,W        ;MDIR-WORK,从原转动方向数据
                               ;作减法运算
        BTFSC    STATUS,Z      ;MDIR=WORK? 若成立
        GOTO     NOCHG         ;不变
CHG     MOVLW    00H           ;W←00000000B (停止)
        MOVWF    PORTB         ;将转动数据输出至端口 B
        CALL     TIMER1        ;切换的时间延迟
NOCHG   MOVF     WORK,0        ;读出转动数据
        RETURN

TIMER1  MOVLW    D62           ;0.1ms 定时器子程序
        MOVWF    CNT1
LOOP1   NOP
        DECFSZ   CNT1,1
        GOTO     LOOP1
        RETURN

TIMER2  MOVLW    D100          ;10ms 定时器子程序
        MOVWF    CNT2;
LOOP2   NOP
        CALL     TIMER1
        DECFSZ   CNT2,1
        GOTO     LOOP2
        RETURN
```

```
TIMER3      MOVLW      D100          ;1s定时器子程序
            MOVWF      CNT3;
LOOP3       NOP
            CALL       TIMER2
            DECFSZ     CNT3,1
            GOTO       LOOP3
            RETURN

            END                      ;程序结束
```

第3章　程序清单3.2

```
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;程序清单3.2
;直流电机简易PWM控制程序
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
            LIST       P = PIC16F84A     ;指定使用的PIC
            INCLUDE    "P16F84A.INC"     ;指定读人设定文件
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
WORK    EQU     0CH                 ;保存转动数据
CNT1    EQU     0DH                 ;定时器1的计数变量
CNT2    EQU     0EH                 ;定时器2的计数变量
CNT3    EQU     0FH                 ;定时器3的计数变量
MDIR    EQU     10H                 ;转动方向的变量
 * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
        ORG     0                   ;存储程序的初始地址

        BSF     STATUS,RP0          ;选择存储单元第1列
        MOVLW   0F0H                ;把11110000B代人W寄存器
        MOVWF   TRISB               ;设定端口B的上位为输入、下位为
                                    ;输出
        BCF     STATUS,RP0          ;选择存储单元第0列
        CLRF    PORTB               ;端口B清零(电机停止)
        MOVLW   01H                 ;W←00000001B (正转)
        MOVWF   MDIR                ;设定原转动方向为正转

 LOOP   MOVLW   01H                 ;W←00000001B (正转)
        BTFSC   PORTB,5             ;检查RB5
```

```
        MOVLW   02H             ;若为 1(ON)W←00000010B(反转)
        MOVWF   WORK            ;保存转动数据
        CALL    CKMDIR          ;检查转动方向
        MOVWF   PORTB           ;将转动数据输出至端口 B

        BTFSS   PORTB,4         ;检查 RB4
        GOTO    LOOP0           ;若为 0(OFF),连续转动

        MOVF    WORK,0          ;若为 1(ON),间歇转动,读出转动数据
        MOVWF   PORTB           ;将转动数据输出至端口 B
        CALL    TIMER2          ;调用 10ms 定时器
        MOVLW   00H             ;W←00000000B(停止)
        MOVWF   PORTB           ;将转动数据输出至端口 B
        CALL    TIMER2          ;调用 10ms 定时器

LOOP0   MOVF    WORK,0          ;读出转动数据
        MOVWF   MDIR            ;保存转动数据
        GOTO    LOOP            ;循环
                                ;转动方向检查子程序
CKMDIR  SUBWF   MDIR,W          ;MDIR-WORK 从原转动方向数据作减法运算
        BTFSC   STATUS,Z        ;MDIR = WORK? 若成立,
        GOTO    NOCHG           ;不变
CHG     MOVLW   00H             ;W←00000000B(停止)
        MOVWF   PORTB           ;向端口 B 输出转动数据
        CALL    TIMER1          ;切换的时间延迟
NOCHG   MOVF    WORK,0          ;读出转动数据
        RETURN

TIMER1  MOVLW   D'62'           ;0.1ms 定时器子程序
        MOVWF   CNT1
LOOP1   NOP
        DECFSZ  CNT1,1
        GOTO    LOOP1
        RETURN

TIMER2  MOVLW   D'100'          ;10ms 定时器子程序
        MOVWF   CNT2
```

```
LOOP2              NOP
        CALL       TIMER1
        DECFSX     CNT2,1
        GOTO       LOOP2
        RETURN

        END                                ;程序结束
```

第 4 章　程序清单 4.1

```
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;程序清单 4.1
;步进电机控制程序(单相励磁)
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
            LIST       P = PIC16F84A        ;指定使用的 PIC
            INCLUDE    "P16F84A.INC"        ;转动初始数据
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
PMD         EQU        08H                  ;
CNT1        EQU        0CH                  ;定时器 1　的计数变量
CNT2        EQU        0DH                  ;定时器 2　的计数变量
CNT3        EQU        0EH                  ;定时器 3　的计数变量
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
            ORG        0                    ;存储程序的初始地址

            BSF        STATUS,RP0           ;选择存储单元第 1 列
            CLRF       TRISB                ;设定端口 B 为输出模式
            BCF        STATUS,RP0           ;选择存储单元第 0 列
            CLRF       PORTB                ;端口 B 清零(电机停止)

NEW         BCF        STATUS,C             ;C 标志位清零
            MOVLW      PMD                  ;设定转动初始数据
            MOVWF      PORTB                ;将转动数据输出至端口 B
MADA        CALL       TIMER3               ;调用定时器
            RRF        PORTB,1              ;端口 B 右移一位
            BTFSS      STATUS,C             ;检查 C 标志位
            GOTO       MADA                 ;若为 0,继续转动
            GOTO       NEW                  ;若为 1,数据初始化
```

```
TIMER1      MOVLW       D'62'           ;0.1ms定时器子程序
            MOVWF       CNT1
LOOP1       NOP
            DECFSZ      CNT1,1
            GOTO        LOOP1
            RETURN

TIMER2      MOVLW       D'100'          ;10ms定时器子程序
            MOVWF       CNT2
LOOP2       NOP
            CALL        TIMER1
            DECFSZ      CNT2,1
            GOTO        LOOP2
            RETURN
TIMER3      MOVLW       D'50'           ;0.5s定时器子程序
            MOVWF       CNT3
LOOP3       NOP
            CALL        TIMER2
            DECFSZ      CNT3,1
            GOTO        LOOP3
            RETURN

            END                         ;程序结束
```

第4章　程序清单4.2

```
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;程序清单4.2
;步进电机控制程序(二相励磁)
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
            LIST        P=PIC16F84A     ;指定使用的PIC
            INCLUDE   "P16F84A.INC"     ;指定读入设定文件
* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
PMD         EQU         4CH             ;转动初始数据
CNT1        EQU         0CH             ;定时器1的计数变量
```

```
CNT2    EQU     0DH          ;定时器 2 的计数变量
CNT3    EQU     0EH          ;定时器 3 的计数变量
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
        ORG     0            ;存储程序的初始地址

        BSF     STATUS,RP0   ;选择存储单元第 1 列
        CLRF    TRISB        ;设定端口 B 为输出模式
        BCF     STATUS,RP0   ;选择存储单元第 0 列
        CLRF    PORTB        ;端口 B 清零(电机停止)

NEW     BCF     STATUS,C     ;C 标志位清零
        MOVLW   PMD          ;设定转动初始数据
        MOVWF   PORTB        ;将转动数据输出至端口 B
MADA    CALL    TIMER3       ;调用定时器
        RRF     PORTB,1      ;端口 B 右移 1 位
        BTFSS   PORTB,7      ;检查第 7 位
        GOTO    MADA         ;若为 0,继续转动
        GOTO    NEW          ;若为 1,数据初始化

TIMER1  MOVLW   D'62'        ;0.1ms 定时器子程序
        MOVWF   CNT1
LOOP1   NOP
        DECFSZ  CNT1,1
        GOTO    LOOP1
        RETURN

TIMER2  MOVLW   D'100'       ;10ms 定时器子程序
        MOVWF   CNT2
LOOP2   NOP
        CALL    TIMER1
        DECFSZ  CNT2,1
        GOTO    LOOP2
        RETURN
TIMER3  MOVLW   D'50'        ;0.5s 定时器子程序
        MOVWF   CNT3
LOOP3   NOP
        CALL    TIMER2
```

```
        DECFSZ      CNT3,1
        GOTO        LOOP3
        RETURN

        END                         ;程序结束
```

第 4 章　程序清单 4.3

```
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;程序清单 4.3
;步进电机控制程序(1-2 相励磁)
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
        LIST        P = PIC16F84A   ;指定使用的 PIC
        INCLUDE  "P16F84A.INC"      ;指定读入设定文件
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
PM1     EQU         08H             ;单相转动数据
PM2     EQU         4CH             ;二相转动数据
PM1W    EQU         0FH             ;单相转动数据保存区
PM2W    EQU         10H             ;二相转动数据保存区
CNT1    EQU         0CH             ;定时器 1 的计数变量
CNT2    EQU         0DH             ;定时器 2 的计数变量
CNT3    EQU         0EH             ;定时器 3 的计数变量
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * *
        ORG         0               ;存储程序的初始地址

        BSF         STATUS,RP0      ;选择存储单元第 1 列
        CLRF        TRISB           ;设定端口 B 为输出模式
        BCF         STATUS,RP0      ;选择存储单元第 0 列
        CLRF        PORTB           ;端口 B 清零(电机停止)

NEW     BCF         STATUS,C        ;C 标志位清零
        MOVLW       PM1             ;设定单相转动的初始数据
        MOVWF       PM1W
        MOVLW       PM2             ;设定二相转动的初始数据
        MOVWF       PM2W

MADA    MOVF PM1W,0                 ;将单相转动数据设置到 W 寄存器中
```

```
        MOVWF       PORTB       ;将转动数据输出至端口B
        CALL        TIMER3      ;调用定时器

        MOVF PM2W,0             ;将二相转动数据设置到W寄存器中
        MOVWF       PORTB       ;将转动数据输出至端口B
        CALL        TIMER3      ;调用定时器

        RRF         RM1W,1      ;单相转动数据右移1位
        RRF         PM2W,1      ;二相转动数据右移1位

        BTFSS       PM2W,7      ;检查第7位
        GOTO MADA               ;若为0,保持原状
        GOTO NEW                ;若为1,从头开始

TIMER1  MOVLW       D62         ;0.1ms定时器子程序
        MOVWF       CNT1

LOOP1   NOP
        DECFSZ      CNT1,1
        GOTO        LOOP1
        RETURN

TIMER2  MOVLW       D100        ;10ms定时器子程序
        MOVWF       CNT2
LOOP2   NOP
        CALL        TIMER1
        DECFSZ      CNT2,1
        GOTO        LOOP2
        RETURN

TIMER3  MOVLW       D50         ;0.5s定时器子程序
        MOVWF       CNT3
LOOP3   NOP
        CALL        TIMER2
        DECFSZ      CNT3,1
        GOTO        LOOP3
        RETURN

        END                     ;程序结束
```

第 7 章　程序清单 7.1

```
;*****************************
;  程序清单 7.1
;  巡线小车控制程序
;*****************************
          LIST     P=PIC16F84A        ;指定所使用的 PIC
          INCLUDE  "P16F84A.INC"      ;指定读取设定文件
;*****************************
          ORG        0                ;程序存储起始地址

          BSF        STATUS,RP0       ;选择存储单元第 1 列
          MOVLW      010H             ;将 00010000B 代入 W 寄存器
          MOVWF      TRISA            ;将端口 A 的上位的 1bit 设输入模式,
                                      ;下位的 4bit 设置输出模式
          MOVLW      071H             ;将 01110001B 代入 W 寄存器
          MOVWF      TRISB            ;将端口 B 的上位 1bit 输入 + 3bit 输
                                      ;出(原文误写做输入)模式,下位的
                                      ;3bit 输出 + 1bit 输入模式设置
          BCF        STATUS,RP0       ;选择存储单元第 0 列
          CLRF       PORTA            ;端口 A 清零(电机停止)
          CLRF       PORTB            ;端口 B 清零

LOOP      CALL       DSP2LED          ;跳转到传感器显示子程序
          BTFSS      PORTA,4          ;检查 RB4
          GOTO       ADJSTOP          ;如果 RB4 = 0 则至停机处理

CKRB6     BTFSS      PORTB,6          ;检查 RB6
          GOTO       MVR              ;如果为 0 则跳转至右转处理
CKRB4     BTFSS      PORTB,4          ;检查 RB4
          GOTO       MVL              ;如果为 0 则跳转至左转处理
CKRB5     BTFSS      PORTB,5          ;检查 RB5
          GOTO       MVS              ;如果为 0 则跳转至前行
          GOTO       ADJSTOP          ;至停止处理

MVR       MOVLW      08H              ;W←00001000B(右转)
          MOVWF      PORTA            ;将右转数据输出至端口 A
          GOTO       LOOP
```

```
MVL         MOVLW     01H          ;W←00000001B(左转)
            MOVWF     PORTA        ;将左转数据输出至端口 A
            GOTO      LOOP
MVS         MOVLW     09H          ;W←00001001B(直行)
            MOVWF     PORTA        ;将前行数据输出至端口 A
            GOTO      LOOP

ADJSTOP     MOVLW     00H          ;W←000000000B(停止)
            MOVWF     PORTA        ;将停止数据输出至端口 A
            GOTO      LOOP
                                   ;传感器数据 LED 指示程序
ADJSTOP     BTFSS     PORTB,4      ;若 RB4 = 0
            BSF       PORTB,1      ;将传感器的检测值输出至端口 B 的第 1 位 LED
            BTFSS     PORTB,5      ;若 RB5 = 0
            BSF       PORTB,2      ;将传感器的检测值输出至端口 B 的第 2 位 LED
            BTFSS     PORTB,6      ;若 RB6 = 0
            BSF       PORTB,3      ;将传感器的检测值输出至端口 B 的第 3 位 LED
            RETURN

            END                    ;程序结束
```

第 7 章　程序清单 7.2

```
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;程序清单 7.2
;巡线小车 PWM 控制程序
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
            LIST       P = PIC16F84A ;指定所使用的 PIC
            INCLUDE   "P16F84A.INC"  ;指定读取设定文件
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
WIDTH       EQU       0C8H           ;设定 LED 熄灭数据
WIDTHP      EQU       64H            ;设定 LED 点亮数据

TMP         EQU       0CH            ;定时器 1 的计数变量
CNT         EQU       0DH            ;定时器 2 的计数变量
DUTYSET     EQU       0EH            ;定时器 3 的计数变量
```

```
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *

            ORG        0               ;程序存储起始地址
            GOTO       MAIN

;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;中断处理程序
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
            ORG        4               ;中断矢量
            MOVWF      TMP             ;W 寄存器保存
            BCF        INTCON,T0IE     ;中断禁止

            MOVF       CNT,W           ;将 CNT 的内容载入 W 寄存器
            SUBLW      WIDTHP          ;WIDTHP-CNT
            BTFSC      STATUS,0        ;CNT< = WIDTHP?
            CALL       DSETFF          ;设定 DUTYSET = FFH

            DECFSZ     CNT,1           ;- - CNT =  = 0?
            GOTO       JMP             ;若 CNT>0 跳转
            MOVLW      WIDTH           ;将 WIDTH 载入 W 寄存器
            MOVWF      CNT             ;将 W 寄存器内容送入 CNT
            MOVLW      00H             ;将 00H 写入 W 寄存器
            MOVWF      DUTYSET         ;设定 DUTYSET 为 00H

JMP         MOVF       TMP,0           ;W 寄存器还原

            RETFIE                     ;中断返回
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;中断处理子程序
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;DUTYSET = 0FF 的计算子程序
    DSETFF  MOVLW      0FFH            ;将 FFH 写入 W 寄存器
            MOVWF      DUTYSET         ;设定 DUTYSET 为 0FFH
            RETURN
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
;主程序
;* * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * * *
```

```
MAIN      BSF      STATUS,RP0       ;选择存储单元第1列
          MOVLW    0D0H             ;定时器0内部无时钟和预定标器
          MOVWF    OPTION_REG.      ;定时器0动作设定

          MOVLW    010H             ;将00010000B代人W寄存器
          MOVWF    TRISA            ;将端口A上位1bit设定为输入模式,
                                    ;下位4bit设定为输出模式
          MOVLW    071H             ;将01110001B代人W寄存器
          MOVWF    TRISB            ;设定端口B上位第7位为输出,6～4
                                    ;位为输入,3～1位为输出,第0位
                                    ;为输入模式
          BCF      STATUS,RP0       ;选择存储单元第0列

          MOVLW    00H              ;将0代人W寄存器
          MOVWF    TMR0             ;向定时器输出
          MOVLW    WIDTH            ;将WIDTH载人W寄存器
          MOVWF    CNT              ;将W寄存器的内容转至CNT

          CLRF     PORTA            ;端口A清零(电机停止)
          CLRF     PORTB            ;端口B清零
          BCF      INTCON,T0IF      ;定时器中断标志=0
          BSF      INTCON,T0IE      ;开定时器0中断许可
          BSF      INTCON,GIE       ;全局中断许可

LOOP      CALL     DSP2LED          ;调用传感器数据显示程序
          BTFSS    PORTA,4          ;检查RB4
          GOTO     ADJSTOP          ;若RB4=0则至停止处理
;检查RB4和RB5是否同时点亮
CKRB45    BTFSS    PORTB,5          ;检查RB5
          GOTO     CKRB4W           ;若为0则继续,且转至检查RB4
          GOTO     CKRB56           ;转至检查RB5和RB6是否同时点亮
CKRB4W    BTFSS    PORTB,4          ;检查RB4
          GOTO     PWML             ;若为0则跳转至PWM左转处理

;检查RB5和RB6是否同时点亮
CKRB56    BTFSS    PORTB,5          ;检查RB5
          GOTO     CKRB6W           ;若为0则继续,且转至检查RB6
          GOTO     CKRB6            ;转至检查RB6点亮
```

```
CKRB6W      BTFSS     PORTB,6        ;检查 RB6
            GOTO      PWMR           ;若为 0 则跳转至 PWM 右转处理

CKRB6       BTFSS     PORTB,6        ;检查 RB6
            GOTO      MVR            ;若为 0 则跳转至右转处理
CKRB4       BTFSS     PORTB,4        ;检查 RB4
            GOTO      MVL            ;若为 0 则跳转至左转处理
CKRB5       BTFSS     PORTB,5        ;检查 RB5
            GOTO      MVS            ;若为 0 则跳转至前行处理
            GOTO      ADJSTOP        ;转至停止处理

MVR         MOVLW     08H            ;W←00001000B(右转)
            MOVWF     PORTA          ;将右转数据输出至端口 A
            GOTO      LOOP
MVL         MOVLW     01H            ;W←00000001B(左转)
            MOVWF     PORTA          ;将左转数据输出至端口 A
            GOTO      LOOP
MVS         MOVLW     09H            ;W←00001001B(前行)
            MOVWF     PORTA          ;将前行数据输出至端口 A
            GOTO      LOOP
PWMR        MOVF      DUTYSET,0      ;W←DUTYSET
            ANDLW     08H            ;W←DUTYSET&00001000B(右转)
            MOVWF     PORTA          ;将右转数据输出至端口 A
            GOTO      LOOP

PWML        MOVF      DUTYSET,0      ;W←DUTYSET
            ANDLW     08H            ;W←DUTYSET&00000001B(左转)
            MOVWF     PORTA          ;将左转数据输出至端口 A
            GOTO      LOOP

ADJSTOP     MOVLW     00H            ;W←000000000B(停止)
            MOVWF     PORTA          ;将停止数据输出至端口 A
            GOTO      LOOP

;传感器数据 LED 指示程序
DSP2LED     BTFSS     PORTB,4        ;若 RB4 = 0
            BSF       PORTB,1        ;将传感器的检测值输出至端口 B 的第 1 位
                                     ;LED
```

```
BTFSS     PORTB,5      ;若 RB5 = 0
BSF       PORTB,2      ;将传感器的检测值输出至端口 B 的第 2 位
                       ;LED
BTFSS     PORTB,6      ;若 RB6 = 0
BSF       PORTB,3      ;将传感器的检测值输出至端口 B 的第 3 位
                       ;LED
RETURN

END                    ;程序结束
```

参考文献

[1] 電子工作のためのPIC活用ガイドブック，技術評論社，後閑哲也

[2] PICマイコンではじめる作って遊べるロボット工作，技術評論社，後閑哲也

[3] 図解PICマイコン実習，森北出版，堀桂太郎

[4] H8マイコン入門，東京電機大学出版局，堀桂太郎

[5] はじめて学ぶC言語マスターブック，オーム社，堀桂太郎

[6] トランジスタ技術　1998年9月号「モータ制御技術の基礎と応用」，CQ出版社

[7] トラ技ORIGINAL　No.7　特集「製作実験 マイコン＆メカトロニクスの誕生」，CQ出版社

[8] 初めて学ぶディジタル回路入門ビギナー教室，オーム社，堀桂太郎
（韓国語，中国語に翻訳出版されています）

[9] はじめてのロボット創造設計，講談社，米田 完他2名共著

[10] 初めて学ぶ電子制御入門早わかり「ロボット力士の作り方」，オーム社，堀桂太郎他
（韓国語，中国語に翻訳出版されています）

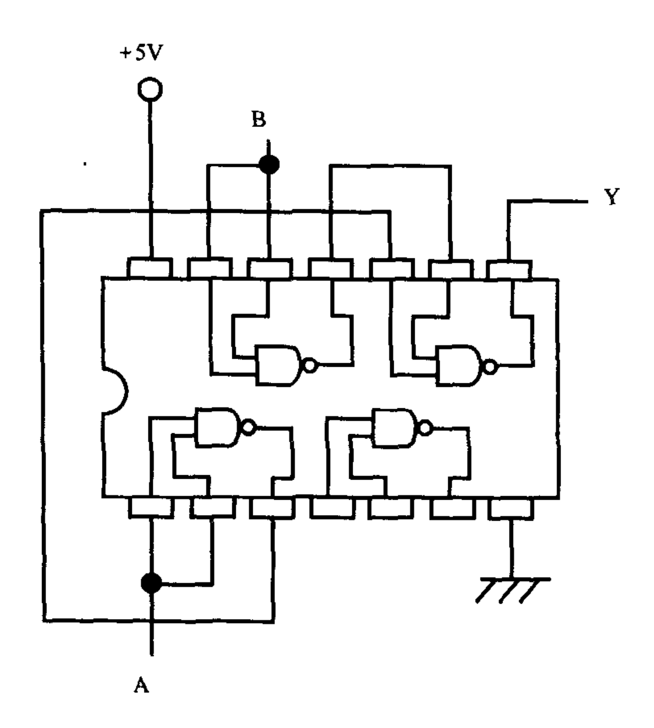
练习题答案

第 1 章

1. 运算、存储及控制
2. ①10　②14　③7　④15　⑤10　⑥13
3. ① 串联总电阻 2kΩ、并联总电阻 500Ω。
   ② 串联总电阻 25kΩ、并联总电阻 6kΩ。
4. ① 根据茶(1)黑(0)茶(1)银(±10%)，因为是 100Ω 的阻值，故电阻范围是 90～100Ω。
   ② 根据橙(3)橙(3)茶(1)金(±5%)，因为是 330Ω 的阻值，故电阻范围是 314～347Ω。
5. ① 串联总电容 0.5μF、并联电容 2μF。
   ② 串联总电容 6μF、并联总电容 25μF。
6. 4 节干电池串联，电压为 6V，电阻的计算值如下：

$$R(\Omega)=\frac{6-2\text{V}}{10\text{mA}}=\frac{4\text{V}}{10\text{mA}}=400\Omega$$

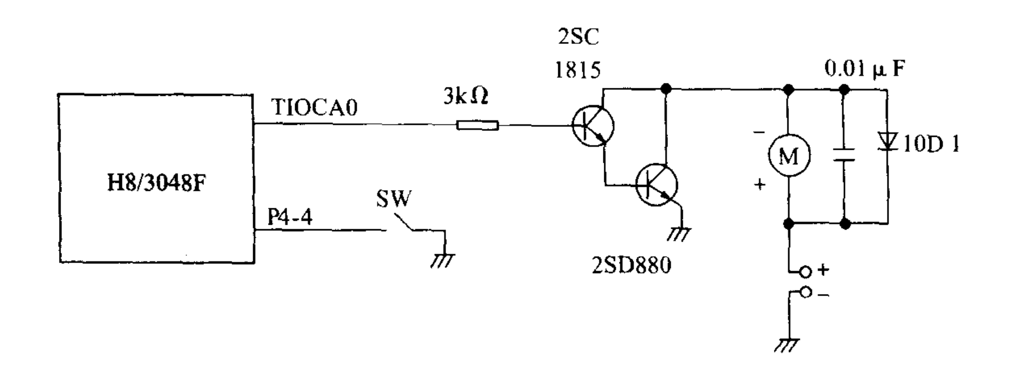
   故应采用 E24 系列的 390Ω 的电阻。
7. ① NPN 型高频用。　② NPN 型低频用。　③ PNP 型高频用
8. 答案见下图



9. · 输入阻抗极大
   · 输出阻抗极小
   · 放大系数接近无穷
   · 从直流至高频均有放大作用

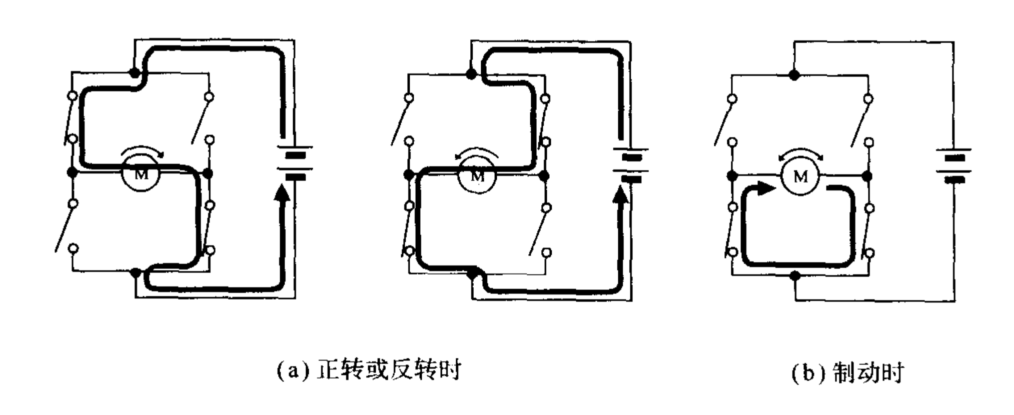
第 2 章

1. 例如，通过学习 PIC 程序编写这样的程序：LED 依次轮换发光，光看起来像流动一样。也可以根据开关的设定让 LED 闪烁变化等。
2. PIC16F84 装有程序存储器，文件寄存器，数据寄存器，栈等 4 种类型的存储器（详细可参见表 2.3）。
3. 在哈佛型的计算机中，储存命令的存储器是与储存数据的存储器分离的（参见 2.2.3 节）。
4. 虽然 H8 具有较大的存储器容量和更强的周边功能，但 PIC 价格便宜，并可以提供免费使用的 MPLAB 开发环境。
5. 个人计算机、PIC 写入器、MPLAB 等（参见 2.2.4 节）。
6. 在 CPU 基础上，把存储器、接口等集成到一起即是微控制器（见图 2.7）。
7.



第 3 章

1. ③错误，没有制动力矩。
2. 在 PWM 控制的 OFF 时间段内，仍有试图使电机转动的电流通过。
3. 转动时整流器的机械触点产生噪声。为了防止噪声，应该在电机的电源两端并联电容器。
4. 正转或反转均为正确答案。



(a) 正转或反转时　　(b) 制动时

5. PWM 控制是一种速度控制方式，就是给直流电机施加高速的 ON/OFF（开关）脉冲信号，改变脉冲信号的 ON/OFF 的比例来进行速度控制。
6. 使用专用 IC 芯片，可在减少元器件的个数，同时可提高电路的可靠性。

7. 增量式编码器以脉冲信号作为角度信号输出，所以只能得到相对角度；而绝对式编码器把角度作为数字信号输出，因此可以得到绝对角度。

第4章

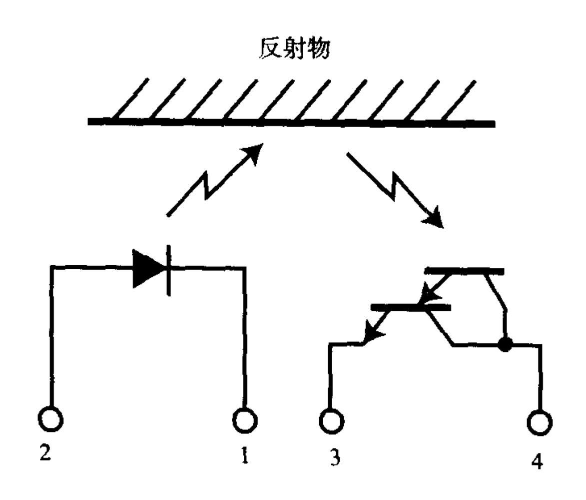
1. ③错误，步进电机不适应于高速转动或大转矩的场合。
2. 由于 90/7.5=12，要转动 90°，就需要使电机转动 12 步。
3. 二相励磁方式。
4. (1)单相励磁方式：电力消耗少。

   (2)二相励磁方式：与单相励磁方式相比，虽然可以得到2倍的转矩，但电力消耗也是其2倍。

   (3)1-2 相励磁方式：是一种单相励磁与二相励磁交替进行的方式，可以得到单相励磁方式的半个步距角，但消耗电力是单相励磁方式的 1.5 倍。
5. 如果按照微控制器的速度使步进电机转动，步进电机跟不上。
6. 例如，只要指定 8 位数据的位数，就可向右或向左移位。
7. 采用专用的 IC 芯片，既可减少元器件的个数，同时又可提高电路的可靠性。
8. 例如以微型机器鼠为例，在探索迷宫时，该功能设法减少流过步进电机的电流以节约电池的电能；而在高速直线前进时，通过增大电流，使步进电机获得大的转矩。

第5章

1. 眼：光电三极管、CCD 图像传感器等；耳：麦克风；鼻：气体传感器，气味传感器；舌：Ph 传感器；皮肤：热敏电阻，触觉传感器。
2. 380～780nm
3. 加速度传感器。车体碰撞时，测定加速度来估计碰撞的强度，以控制安全气囊的动作。
4. 20Hz～20kHz
5. $V=331.5+0.6\times0=331.5(\mathrm{m/s})$； $V=331.5+0.6\times50=361.5(\mathrm{m/s})$
6. $R(\Omega)=\dfrac{5-2V}{10\mathrm{mA}}=\dfrac{3V}{10\mathrm{mA}}=300\Omega$
7. 来自 LED 的辐射光，被反射物体反射后到达光电三极管，由此来检测反射物体。



第6章

1. 波长超过770nm的眼睛看不见的光。
2. ①正向 ②电流 ③无法 ④能
3. ①调制 ②40kHz ③4万 ④600μs
4. 对红外线进行调制时，是根据微控制器的时钟频率计算周期的。因此若改变时钟频率，就必须对程序加以修改。
5. 开始位的作用是告知接收器开始向它发送数据。而停止位的作用是接收器确认接收到的数据是否为应该接收的数据。
6. 适当设置发送器和接收器的方向。在可能操作距离内使用。发送器和接收器之间不要放置物体遮挡红外线。
7. ①未按下按钮时，利用PIC微控制器内部的上拉电路，可以让PIC微控制器的输入引脚(RB0～RB4)置为“1”电平。

   ②设置20μs的停止发送时间作用是避免连续操作按钮时，接收器出现误动作。

   ③接收到红外线则输出“0”电平，未接收到红外线则输出“1”电平。

   ④即使只按一次按钮也连续发送两次动作数据，接收侧将动作数据进行比较。如果添加了只有当前后两次数据一致时才认为该动作数据有效的功能，显然就可以提高可靠性。如果增加停止位则可以进一步提高可靠性。

第7章

1. PIC16F84A(20MHz)的电压为4.5～5.5V(随频率和振荡器的不同而异)。TA7257P有V_{cc}为6～18V和V_s为0～18V两种。因此，不行。
2. 前进、后退、右转、左转、停止、以及左右旋转动作共有7个命令，因此动作数据最少需要3位(共有$2^3=8$种表达法)才行。另外，向TA7257P发送1001、0110两组动作数据。
3. 电机端子上的电容器能吸收交流成分的噪声。如果电容器安装得过远，导线中仍会漏出一些噪声，所以应尽可能地靠近电机接线端子。
4. 减速比为203.7∶1时转矩较大。
5. 钻孔时如用力过猛容易碎裂，请注意。使用丙烯专用切断刀时应均匀用力。
6. 从光电判读器内部的LED发出的光照射到黑线上时，反射光的强度会减小从而检测出黑线来。

索　引

(TP-2451.0101)
责任编辑 杨 凯 崔炳哲
责任制作 魏 谨
责任印制 刘士平
封面设计 李 祥



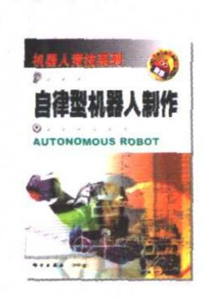
机器人创意与制作

- 机器人控制电子学
- 机器人设计与控制
- 机器人控制器与程序设计
- 小型机器人的基础技术与制作

机器人相关系列图书



















科学出版社 东方科龙
www.okbook.com.cn



ISBN7-03-013168-1
定 价：28.00元